[redacted]

DEVELOPMENT OF A DIFFERENTIAL GPS TRACKING SYSTEM

FOR SOUNDING ROCKET PAYLOADS

by

Jay Helmericks

RECOMMENDED:

John Aspnes

Joseph S. Hawkins
Advisory Committee Chair

John Aspnes for CEM
Department Head

APPROVED:

Dean, College of Science Engineering and Mathematics

Dean of the Graduate School

8-3-01
Date

DEVELOPMENT OF A DIFFERENTIAL GPS TRACKING SYSTEM

FOR SOUNDING ROCKET PAYLOADS

A

THESIS

Presented to the Faculty

of the University of Alaska Fairbanks

in Partial Fulfillment of the Requirements

for the Degree of

MASTER OF SCIENCE

By

Jay Gregory Helmericks, B.S.

Fairbanks, Alaska

August 2001

Abstract

The purpose of this thesis was to develop a system that could track a sounding rocket payload with a commercial GPS receiver. A GPS receiver was chosen that still outputs raw data when the COCOM limits are exceeded. All the hardware to support the OEM GPS receiver in a reverse differential system was designed and built, including both a ground system and two flight systems to support both on-board storage and telemetry. A software program was developed to archive and compute positions from the raw data. The GPS system has been ground tested and flown on an Orion sounding rocket. The testing shows that the system works and the expected accuracy is 10-50 ft depending on the distance between the ground station and the rocket, satellite geometry and other sources of error.

Table of Contents

List of Figures

List of Tables

Acknowledgments

I would like to thank Dr. Hawkins and the Alaska Space Grant Program for providing the funding for this thesis, both for the hardware and the research assistantship that kept me fed and housed. I thank Randy Thomas for his help in launching a GPS payload on his hybrid rocket.

I thank my wife Amy, the most awesome, loving, fabulously beautiful woman on the planet, for encouraging me, believing that I could finish this, and helping with proof reading and grammar flow. I also thank her for cooking and making me get some sleep while writing this thesis. A huge debt of gratitude goes to my mother and father-in-law who removed every obstacle they could, going so far as to fix my car when it broke, so I could continue working. I thank my Mom and Dad for making this possible and encouraging me to go after my dreams.

Chapter 1: Introduction and Overview

The goal of this thesis was to develop a reverse Differential Global Positioning System (DGPS) tracking system for future Alaska Student Rocket Program (ASRP) payloads. The ASRP is sponsored by the Alaska Space Grant Program (ASGP) at the University of Alaska Fairbanks. Students participating in the ASRP design and build sounding rockets that are launched at the nearby Poker Flat Research Range with the assistance of the National Aeronautics and Space Administration (NASA). The students design and build the entire rocket except for the motor, which is provided by NASA.

One of the range safety requirements for launching a sounding rocket is that it must be possible to track the path of the rocket. NASA keeps a record of each flight and uses this information to predict the nominal dispersion of the expected flight path and downrange impact area for future launches. The other use of tracking information is for payload recovery. Knowing the position and flight path as the payload drops over the horizon greatly reduces the search area.

Rocket tracking is traditionally done with radar, but bringing a NASA radar team to Fairbanks for a flight is expensive. For this reason it was decided to develop a self-tracking rocket using a GPS receiver. The rocket is called self-tracking because it transmits its position to the ground during flight instead of relying on radar tracking. Once the GPS system is proven reliable it will become a part of the standard housekeeping electronics on all ASRP payloads. In the future, this might permit rockets to be launched without the need for a radar tracking team.

NASA and the military have already proven that it is possible for high dynamic GPS receivers to track a rocket. However, these robust receivers cost $10,000 to $100,000 each. The objective of this thesis was to develop a system around a civilian GPS receiver costing only $150 and evaluate its accuracy. The main limitation of using civilian receivers is that U.S. export regulations limit the speed and altitude that the GPS receivers are allowed to function. Fortunately, the Rockwell Jupiter receiver, continues to output raw data and only cuts off the calculated position when the limits are exceeded. The GPS tracking system presented here is based on the use of this raw data, which is expected to remain valid throughout the rocket flight. A Differential GPS (DGPS) system was chosen to increase the accuracy by using two receivers, one of which generates corrections that the other receiver uses. Chapter 2 covers the GPS system in more detail.

The hardware support needed for the commercial GPS receiver required the development of several components. The hardware was designed to support rockets with and without telemetry. The storage system makes it possible to test the GPS system in small amateur rockets before major missions. Chapter 3 details the different hardware components that were developed.

A program was developed to compute the position from the raw data output by the GPS receiver. The program is used both to calculate the position of the flight receiver and archive the data. An embedded microcontroller is used in the storage board and a program was developed for it. For the full details of the software programs see chapter 4.

The performance of the GPS system was evaluated on the ground and in flight. Chapter 5 covers the ground based testing. Stationary testing is examined first because the antenna position was known (within a couple of feet) and it was easy to estimate the error that was present in the measurement. Next, the receiver's calculated output was examined with different settings to evaluate the receiver performance. This chapter then examines the program's position calculation performance. Finally, the moving performance of both the receiver and the program are examined with a data set taken in a car. The actual position is no longer known, so only a comparison between the methods of calculation is examined.

Chapter 6 examines the flight data the GPS system collected during the launch of a NASA Orion rocket. The data is very interesting to analyze since this is the same motor that NASA currently provides to ASRP. Unfortunately the data from this flight was corrupted making it unprocessable by the positioning program. However, significant conclusions about the GPS system's performance are still achieved.

Conclusions in chapter 7 briefly highlight the main developments and results.

Chapter 2: Brief overview of the GPS System

The Global Positioning System is designed to allow users to find their location anywhere on the surface of the earth. To achieve this, a minimum of 21 satellites orbit the earth in 6 orbital planes inclined at 55° and spaced about 60° apart. A couple of in-orbit spares are maintained, so the constellation is usually about 24 satellites. They are in 11 h 58 min circular orbits with a radius of 20,000 km. This makes the constellation as seen at a stationary point on the earth repeat itself twice a day. A view taken at the same time each day will show a slight progression as the orbits are just faster than half a day. The number and spacing of the satellites was designed to provide at least four satellites above the horizon anywhere on the earth, 24 hours a day. The high latitude of Alaska increases the number of satellites that are normally visible, but causes them to be lower to the horizon. Twelve satellites are possible, and it is common to track eight. The lower satellite elevation increases the atomospheric errors so the increased number of satellites does not necessarily improve the position solution.

The satellites transmit on two frequencies, 1.57542 GHz (L1) and 1.22760 GHz (L2). L1 is a general frequency that all users can access while L2 is reserved for military use. Two codes (C/A, and P) and a 50 bps data stream are modulated onto L1, and one code (P) and the data stream are modulated onto L2.

The C/A code is called the coarse/acquisition code, and this is the only code available to the general public. The C/A code is a pseudorandom bit stream that is transmitted at 1.023 MHz. The code is 1023 bits long, so the code repeats every millisecond. There are 32 possible codes and each satellite transmits a different one. The code is used to identify and distinguish the different satellites since they all transmit on the same frequency. The transmission rate of the C/A code results in a code wavelength of 300 meters. This long wavelength is why it is called the coarse code. The accuracy of the position is limited by the wavelength of signal being measured. The C/A code is only modulated onto the L1 carrier. Because of this, civilian receivers are limited to single frequency operation.

The P code is called the precision code because it is transmitted at 10.23 MHz. This lowers the code wavelength to 30 meters, enhancing the inherent accuracy of the signal. Unfortunately for general users, the P code is encrypted to the Y code, so that only the military or other approved users can access it. The P code is a very long pseudorandom sequence and it takes almost 267 days to repeat. Because of the long sequence it is very hard to lock on from scratch, which is why the C/A code is called the acquisition code. Once the C/A code is tracked, the data stream can be demodulated and a hand over word helps the receiver lock onto the P code by indicating a point in the sequence to start looking. The P code (really the Y code because of the encryption) is modulated onto both L1 and L2. Users that have access to the P code can use dual frequency receivers that remove the ionospheric delay, improving the positioning accuracy.

The data stream contains the satellite ephemeris, correction data, and the almanac. The data set is repeated every 12 minutes. The ephemeris contains very accurate orbital parameters and is used to calculate the satellite's position. The ephemeris is good for about four hours, but for maximum accuracy it is updated every hour by ground control. The correction information is used to correct for satellite clock errors and atmospheric effects. The almanac contains orbital parameters on all the satellites and is accurate for weeks. It is used to calculate rough satellite positions on initial startup to speed up the search for satellites. Kaplan provides all of the previous information in more detail [Kaplan].

The GPS receiver tracks the satellite signal by generating a replica of the code. The replica is then time shifted until the right time delay is found that demodulates the signal. As long as the code delay is incorrect, the information is lost in the noise. But when the internal code lines up with the received code, it lifts the data out of the noise, allowing the signal to be tracked. The receiver then measures the time difference between the time in the decoded information, which is when the signal was sent, and the time in the receiver when the signal was received. The difference is multiplied by the speed of light to get the range from the satellite to the receiver. The receiver uses an inexpensive crystal clock, so there is a large clock error in the measurement. This error will be common to all signals received at the same time, so it represents an additional unknown. Solving for four unknowns (x, y, z, time offset) requires a minimum of four satellites in simultaneous view. Using a fourth satellite to correct for time makes GPS receivers available at low cost, since a large, expensive atomic clock is not needed. One measurement results in a sphere of possible locations around the satellite. Two measurements result in a circle of possible positions where the two spheres intersect. A third satellite results in three range spheres that ideally intersect at two points, one that is right and one that is discarded because it is in an unrealistic location. Because of the clock error, the spheres do not intersect at a point, until a fourth satellite is added to solve for

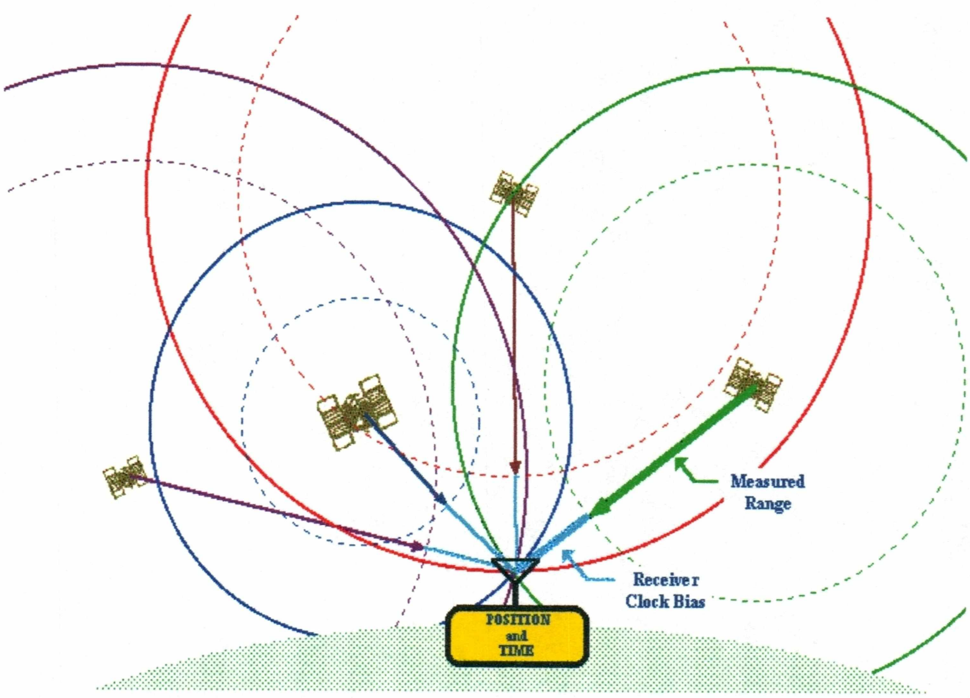

Figure 1: Satellite range intersection

the receiver clock offset. Figure 1, taken from Peter Dana's GPS website, illustrates how this works [Dana]. If more then four satellites are available the measurements can be combined in a least squares solution that minimizes the effect of other error sources.

There are three error categories in the GPS solution: satellite errors, propagation errors, and receiver errors. Satellite errors include the satellite clock offset, position errors resulting from inaccuracies in the ephemeris, and phase errors resulting from the satellite antenna phase center drifting. Propagation errors are caused by the signal not traveling at the speed of light through the ionosphere and troposphere. Dual frequency receivers can remove the ionospheric delay because it is proportional to frequency. By knowing the delay difference between the two frequencies, and the transmission frequencies, the total delay can be determined. Receiver errors include uncorrected clock offset and measurement errors.

Differential GPS (DGPS) attempts to correct the satellite and propagation errors by using two receivers. One receiver, referred to as the ground station receiver, is placed at a known location. The information received from the satellites is then corrected using the known location as a reference. The correction information generated by the ground station receiver is then sent to the second receiver, referred to as the rover or flight receiver, to correct the ranging information that it has measured. The correction information allows the flight receiver to remove some error factors and calculate a much more accurate position. If the two receivers are close together, then the signals will travel the same path, and the satellite and propagation errors will be very similar. As the flight and ground receivers get farther apart the corrections are less and less effective, so that the positioning error slowly increases. This degradation is caused by the difference in the signal propagation path.

Most rocket flights have downlink only or no telemetry system at all. Because of this, the standard DGPS method of sending corrections to the flight receiver is not convenient. For this reason, a reverse DGPS system was designed for ASRP payloads.

In this system the raw data from the flight receiver is transmitted to the ground, where it is processed along with the ground station receiver's data.

The maximum speed and altitude at which the GPS receiver will operate is hardware limited by how much Doppler shift can be tolerated. Many receivers are designed so that doppler shift does not become an issue until near orbital speeds. So hardware is not really a limiting factor for non-military rocket users. However, the U.S. government Coordinating Committee for Multilateral Export Control (COCOM) limits receiver operation to 60,000 feet and 1000 knots in an attempt to prevent GPS receivers from being used to guide rockets and missiles [Gilbert]. This limitation is done in software since the hardware does not have a clean operational limit. Most manufactures voluntarily conform to the guidelines to avoid export restrictions.

Chapter 3: Hardware Implementation

3.1 Hardware System Overview

The DGPS system is made up of two main components: the flight system and the ground system. The flight system includes all the GPS components on the rocket. The ground system includes the ground station GPS and the support computer running the archiving and tracking software.

For differential positioning to work, the data from both GPS receivers must be combined and processed by the software. There are two different ways that this can take place. The first method is a real-time setup that requires current GPS data coming in from both receivers on separate serial ports. The software uses this data to calculate the position of the flight GPS. If the flight GPS is in a distant location, either on the ground or in a rocket, an RF telemetry link is required to get the GPS data to the computer. The second method is post-processing. In this case the flight and ground GPS data are saved to files. The files are then processed by the program at a later time to calculate the position of the flight GPS. For rocket payloads without a telemetry downlink, an on-board storage system must be added to the flight GPS receiver to capture the data for post-flight processing.

3.2 Telemetry Flight GPS System

The telemetry flight system consists of the GPS receiver, the telemetry flight board, and the GPS patch antenna. The telemetry flight board holds the GPS receiver and the lithium backup battery. The data from the receiver is output to the flight computer with a serial connection. ASRP payloads use a standardized electrical bus and mounting configuration [Burket]. The flight board converts the GPS receiver to the ASRP standard connectors and mounting holes. The patch antenna wraps around the outer payload skin so that all the satellites remain in view as the rocket spins.

3.2.1 Telemetry Flight Board

The board used for ASRP sounding rockets is uncomplicated because the standard payload electronics supply regulated power and data management. The board provides the lithium battery needed to backup the memory and real time clock (RTC) on the receiver, and a transition to the mounting and connection standards of ASRP payloads [Burket]. Figure 2 shows a block diagram of the board, and Figure 53 in appendix A shows a detailed schematic of the board.

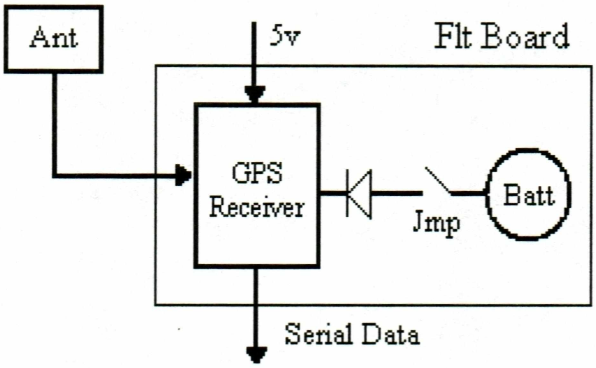

Figure 2: Telemetry Flight Board Block Diagram

The flight board is the standard ASRP 1x2 unit size with the corresponding mounting pattern. The receiver mounts to the flight board and is electrically connected through a 20-pin 2 mm female header. Table 1 lists the connections to the flight board. The data connection goes to a serial port on the flight computer. The power connection goes to the power bus. The receiver has an MCX connector for the antenna cable.

Table 1: Telemetry Flight Board Connections

Type	Name	Pinout
4-pin molex header	std power	±15 V, 5 V, gnd
4-pin molex header	data	gnd, 5 V, serial data in, serial data out

The serial data stream output from the receiver is a 0-5 V logic level output that does not conform to RS-232. For this reason, the data connection is designed to provide the full pinout for the logic level to RS-232 converter to facilitate testing of the board and receiver separate from the rest of the payload. Only two of the pins in the data connection, ground and serial data out, are used by the flight computer.

One jumper connects the lithium battery to the receiver. This allows the battery to be disconnected during storage of the board. This jumper should be connected before flight with a snug jumper cap.

3.3 Storage Flight GPS System

The storage flight GPS system can fly on any rocket without telemetry. Most of the time this setup will be used in small amateur rockets, but it can also be flown on any sounding rocket that does not have telemetry support. Figure 3 is block diagram showing how the electronics fit together. The storage system consists of the GPS receiver, storage flight board, battery pack, and GPS antenna. If two or more antennas are used, then a power combiner is added to the list. The flight board for the storage system is much more complex than that for the telemetry system because it includes all necessary electrical and data handling support. Since small rockets do not travel very high, it is not critical that the ground station be at the launch location. As long as the flight is within 10 to 20 km of

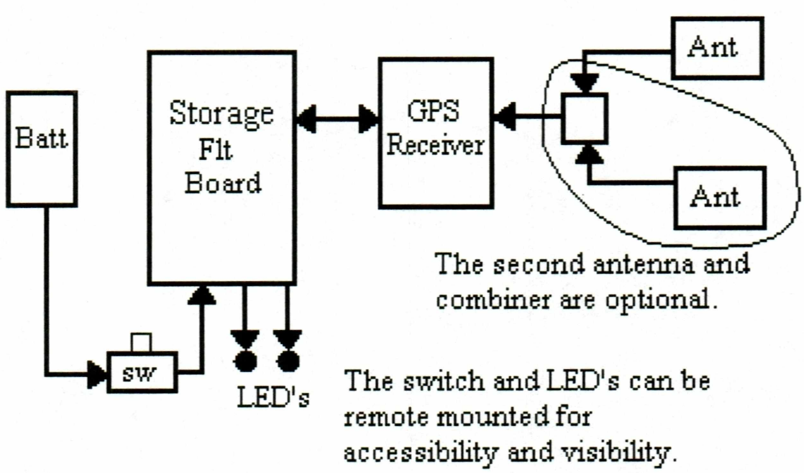

Figure 3: Storage Flight GPS System Block Diagram

the ground station it is not necessary to move it. This is because the signals will still travel roughly the same path due to the high orbit of the satellites.

Two antennas are needed in order to get continuous coverage as the payload rotates and spins during flight. A single antenna pointing up could handle the spin without any problem, until the parachute is deployed, when the payload and the antenna are pointed earthward. If a second antenna is added pointing down initially, the flip over problem is solved, but a very obvious potential for multipath exists with one antenna pointed at the ground all the time. Another option is to mount two antennas pointing sideways, which solves the flip over problem, however testing has shown that it performs poorly when spinning. (See discussion in Section 6.2)

If the storage system is to be flown on an amateur rocket, a complete payload module, which includes the flight system and structure, must be constructed. The equipment is integrated into a payload module to fit the rocket that is being used. A four inch payload module has been constructed for testing purposes, as shown in Figure 4. The module is sitting upside down on the table. This payload is designed to fly in a four inch rocket constructed by Randy Thomas at the University of Alaska Fairbanks. This design has two side facing antennas, which necessitated the double deck layout. The advantage of this layout is increased mounting space for components, but mounting on the inner side limits access, and components on the outer side must be relatively flat near the sides as the headroom is reduced by the curve of the payload tube. The GPS antenna and storage flight board determine the

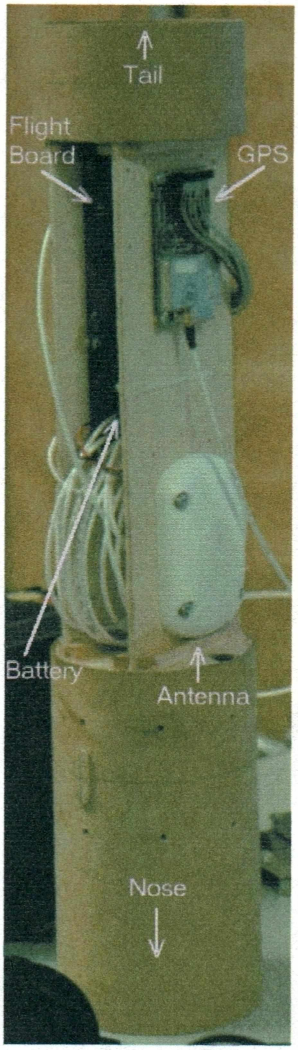

Figure 4: Payload Module with Storage Flight GPS System

minimum length of the payload section, and the power combiner, receiver, and battery were fitted in where possible. The picture shows the receiver and one of the antennas mounted on the outside of one deck. The battery pack is held onto the inner side with tie straps that can be seen between the receiver and antenna. The inner side of the opposing deck holds the flight board and the outer side has the other antenna and the power combiner.

The balsa wood is soft and screws can be easily tapped into it, but the holding strength is limited so it is recommended that nuts be used, or the hole soaked in a liquid glue like CA from Tower Hobbies. Small standoffs were used for the storage and receiver boards so that they could be screwed down without flexing the boards over the components mounted on the underside. The standoffs were glued to the board after mounting the first time to ease reassembly. The GPS receiver must be over a foot away from the antennas or heavily shielded to eliminate feedback that increases the noise floor in the RF section. Due to space limitations the receiver was shielded with four layers of aluminum tape after the photograph in Figure 4 was taken.

3.3.1 Storage Flight Board

The storage flight board was designed to fly in small amateur rockets that could be used to test the GPS system. This resulted in a complex board because many functions were integrated into one board. The board contains power regulation, a microcontroller, and memory with battery backup. Figure 5 shows a block diagram of the board, and Figure 54 in appendix A shows a detailed schematic of the board.

The board was made with a 1.95" width so it could fit into a 2" diameter rocket. The resulting length was 5.5". Although this is a long board, it can be flown in just about any rocket without modification. The length of the board required six mounting holes to prevent the board from bending during high G loads. The extra two holes greatly

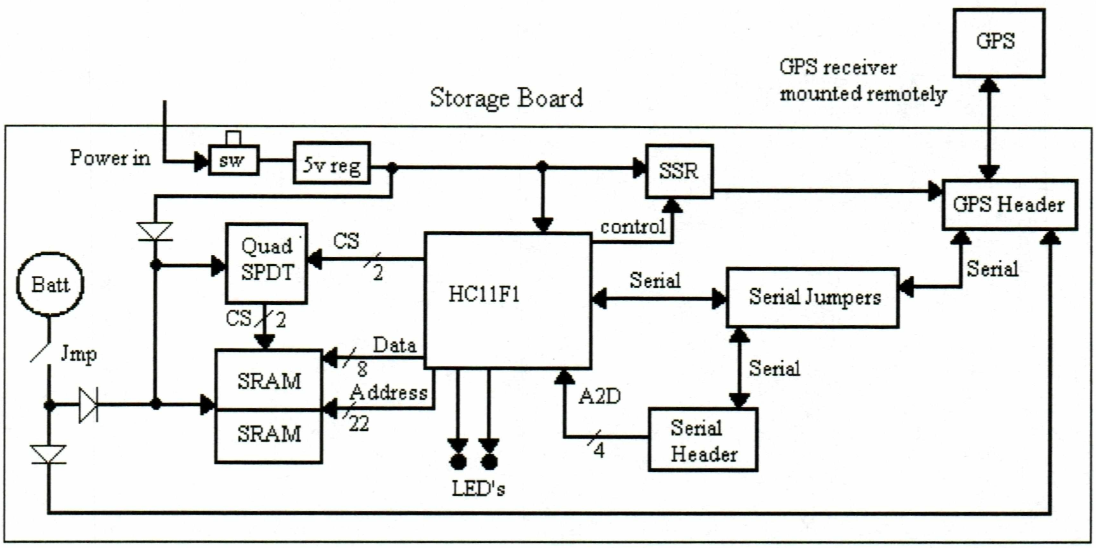

Figure 5: Storage Flight Board Block Diagram

increased the routing difficulty, since they occur in the high density zone near the memory chips.

The main power requirement for the board is 5 V and it is provided by a low dropout regulator from National Semiconductor. The low dropout regulator maintains a useful 5 V output (4.9 V) with an input of 5.25 V. The maximum voltage is constrained by the power dissipation of the regulator. At room temperature, this is around 1.5 watts resulting in a maximum input of 10-11 volts. A low dropout regulator was chosen because it allows the maximum range of useful battery voltages. The regulator requires an output capacitor with an Equivalent Series Resistance (ESR) between 0.1 and 1 Ω to remain stable. The regulator was not stable with a standard metalized film capacitor, but a 22 μF tantalum capacitor was used successfully. The board has been tested down to 0 °F without problems.

The power for the GPS receiver is controlled by a Solid State Relay (SSR) allowing the receiver to be turned on and off by the HC11 microcontroller. The receiver uses most of the power consumed by the electronics, so turning it off allows the board to go into a low power mode. This mode is useful to save the batteries if the rocket is difficult to find after launch. The HC11 monitors the input voltage and shuts down gracefully when the batteries reach the end of their life.

The data from the GPS is stored in SRAM chips. SRAM was used because it makes reading and writing to the chip easy, and simplifies the HC11 program. The disadvantage is that it requires external hardware to backup the memory. Two 4-Mbit chips can store 28 minutes of data. If more time is required, up to four chips can be used on a bigger board with only minor modification to the hardware and software.

A lithium battery is used to backup the SRAM for data storage as well as the SRAM and RTC on the GPS receiver. Maintaining the RTC and SRAM allows the receiver to lockup much faster after a power down. The SRAM holds the ephemeris information and saving this allows the receiver to start navigating as soon as it reacquires the satellites. Saving the RTC allows the receiver to calculate a specific place to start looking instead of blindly searching until the first satellite is found. A jumper allows the battery to be disconnected if the board will be stored for a long time.

Adding the battery backup to the SRAM chips turned out to be challenging because power has to be applied to both the main power and chip select pins. Power has to be provided to the chip select pins because they are active low, so if they are allowed to sink to ground the chip will try to turn on using the power from the battery. The difficulty in doing this is that the HC11 has to have control of the chip selects while it is operating. To do this a quad single pole, dual throw, normally open/normally closed (SPDT NO/NC) switch was used with the control connected to main power. Then, as power is turned on, the switches flip connecting the HC11 to the chip select lines. As power is turned off, the switches flip back and connect the battery to the lines. The chip has internal logic that has to remain powered so it has a diode switch between main power and the battery so it always remains on. An identical diode switch controls the main power of the SRAM chips. A quad switch was used so that there was a built-in capability to expand the number of memory chips.

To provide for future expansion and to increase the usefulness of the board, four of the 8 bit A2D lines from the HC11 were routed to the serial header connector. This provides built-in hardware support, so that only a software change is necessary to sample external information. Any external data can be sampled and stored with the GPS data in RAM.

The HC11 is the heart of the board. It controls the powering of the GPS and the storing and dumping of the data. The chip is a Motorola 68HC11F1 which has more ports than the standard 68HC11E9. The extra ports are needed to address the 8-Mbits of external memory without multiplexing the address and data buses. The HC11 only has a 16-bit address bus so the memory is invisible to the general operation of the HC11. Address and data are accessed through writes and reads to the different ports of the HC11. This is done semi-transparent to the main program through subroutines that handle the reading and writing operations. The F1 has 1 kbyte of RAM and 512 bytes of E^2PROM. Most of the RAM goes unused, since the program is in the E^2PROM and all the data is stored to the SRAM chips. If later programs become too complex to fit internally then an external E^2PROM chip could be used and the code loaded into RAM upon startup.

Table 2: Storage Flight Board Connections

Type	Name
2-pin molex MicroFit 3 (MF3)	Battery Power In
male header 20-pin	GPS connection
shrouded header 10-pin	Serial, A2D connector

Table 2 lists the connections to the flight board. The GPS receiver and the flight board are mounted separately in the payload section and a 20-pin 2 mm cable connects the two together. As long as the cable is connected, battery power is provided to the GPS receiver to backup its memory and RTC.

The serial connection between HC11, GPS and Computer is controlled with a set of six jumpers. See Figure 6 for the layout of the jumpers. The set of jumpers allows the serial link to be routed in any manner between the three devices. The computer can be connected to the GPS or HC11 for setup, testing, and programming, and the GPS and HC11 can be connected together for flight. The most convenient setup for general operations is to connect GPS out to HC11 in, and HC11 out to Computer in. This allows the board to collect data from the GPS and download it to the computer without changing the jumper settings. This works fine as long as the HC11 doesn't need to be programed. Another useful setup during testing is to send the GPS output to both the HC11 and the computer. This allows the data to be monitored while at the same time letting the HC11 process it.

Three jumpers control the function of the HC11. See Figure 6 for the layout of the jumpers. The HC11 is reset by jumper C. Leave this jumper open for normal operation. Jumper B puts the HC11 into the special boot mode. In this mode it waits for the computer to download a bootloader that then takes control of the HC11. The bootloader is downloaded by PCBug, which is explained in Section 4.3. Leave jumper B open for normal operation. Jumper A sets the operating mode of HC11. If open the HC11 will go into store mode and save the GPS data to memory. If the jumper is closed, the HC11 will go into dump mode and download the contents of memory out the serial port. If the board has memory that needs

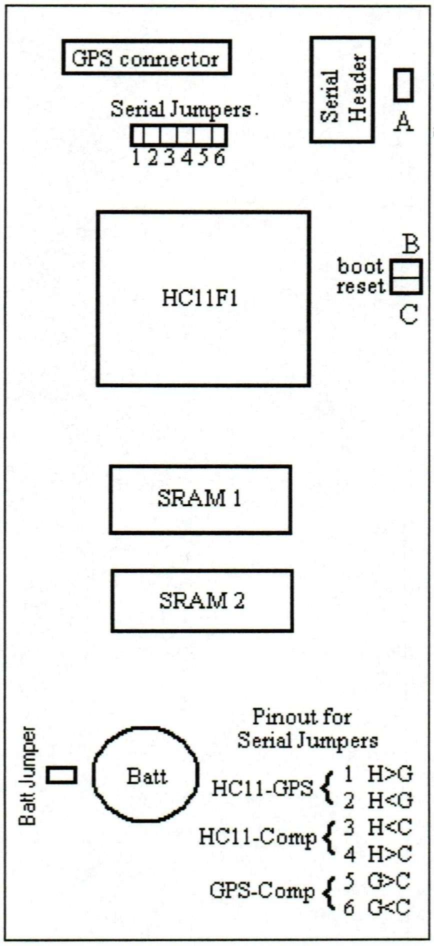

Figure 6: Storage Flight Board Jumper Layout

to be saved this jumper should be closed. All of the HC11 jumpers should be open before flight.

An additional jumper connects the lithium battery to the electronics. This allows the battery to be disconnected during storage of the board. This jumper should be closed before flight.

3.4 Ground GPS System

The ground GPS system is usually located at the telemetry ground station for the rocket. AC power is required for the computer that archives or displays the data and the ground GPS board. The ground GPS antenna needs to be located accurately. This can be accomplished by placing the antenna at a surveyed location or by using the software to determine the location. The software determination averages the position over time, since the errors are generally random and will average to zero. One to two hours is sufficient for nominal accuracy, but six or more hours yield better accuracy. The antenna location is used by the program with the data from the ground GPS to generate the differential corrections, so that any error in the antenna position translates to a equal offset error in the resulting position. For example, if the antenna position error is two feet to the west, then all the position calculations will be offset two feet to the west.

3.4.1 Ground Station Board

The ground station board serves as the necessary interface between the GPS, external power, and the computer. Figure 7 shows a block diagram of the layout and Figure 55 in Appendix A shows a detailed schematic. The board has a 5 V regulator so that any DC supply from 6.5 to 14 V can be used. The power connector is a round power jack so that standard AC/DC converters can be used. The board is then able to be powered from any AC outlet, simplifying use. Currently a 9 V AC/DC converter powers the prototype board.

The other major part of the board is the logic level to RS-232 serial converter. This is a Motorola chip that takes the 0 to 5 V serial stream from the GPS receiver and

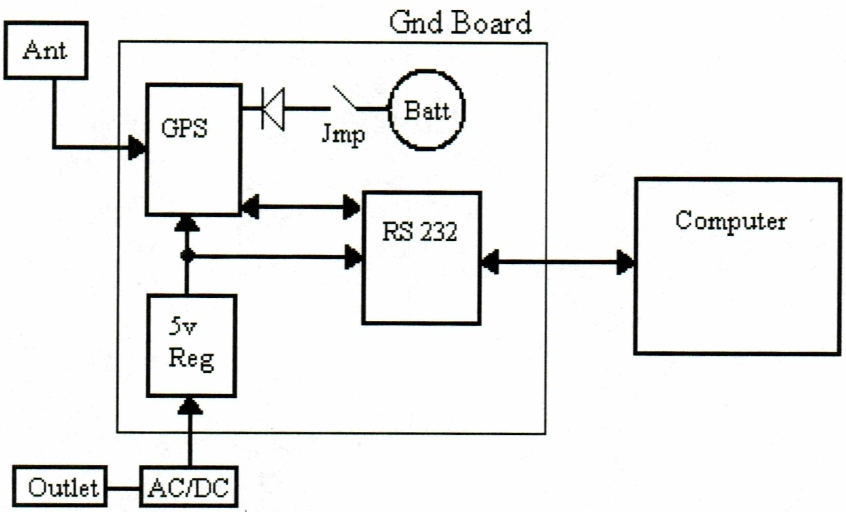

Figure 7: Ground Board Block Diagram

converts it to RS-232. It provides the necessary interface to a computer to collect the data. A lithium battery is used to backup the SRAM and RTC. It can be disconnected if the board is going to be stored for a long time.

Table 3 lists the connections to the ground board. Two jumpers control the GPS receiver. One is the reset jumper and closing it will reset the GPS receiver. The second jumper is a mode select and closing it will select default serial parameters to be used for communication. The default values are 9600 baud, 8 data bits, no parity, and 1 stop bit.

Table 3: Ground Board Connections

Type	Name
Power Jack	Power input
DB9 Male	Serial data out

3.5 Rockwell OEM GPS Receiver

The GPS receiver is an OEM Rockwell Jupiter. The board is based on their Zodiac chipset. It is a 12-channel receiver and tracks all satellites in parallel. All the channels are used together to look for the same satellite on startup for the fastest startup possible. The board uses about 220 mA at 5 volts. For full specifications of the board see the data sheet in the Rockwell Designer's Guide [Rockwell].

The Rockwell receiver was chosen because it is reported to have limited software lockouts. U.S. export regulations limit commercial GPS receiver operation to under 60,000 ft and 1000 knots. This limit was designed to prevent a receiver from being used to guide a missile. Most receivers stop outputting data when the limit is exceeded. The Jupiter, however, doesn't completely cut off, it only stops calculating a position internally and continues to output the raw measurement data. This characteristic is what makes tracking a sounding rocket possible with an inexpensive commercial receiver. By calculating the receiver's position from the raw data a position track can be maintained when the limits are exceeded.

The GPS receiver has configuration options that change its operation and output. The configuration is stored in EEPROM and once set does not need to be reset until changed. The options can be set using the Labmon program from Rockwell. A description of how to use the program is in the Rockwell Designer's Guide [Rockwell].

The settings change how the receiver calculates its internal positioning solution and do not affect the raw data. Pertinent options are shown in Table 4 below. The options shown have been set to maximize the potential quality of the positions that the receiver outputs during rocket flights. For example, the dynamic setting has been set to 'high', and the altitude hold has been turned 'off', since a rocket moves in a largely vertical direction.

Table 4: Receiver Options

Setting	Options	Description	Set to
Elevation Mask	0-90°	Sets the elevation above the horizon that a satellite has to be before the receiver will use it in the position calculation.	10°
Expected Dynamics	high, med, low	Sets the expected level of movement the receiver will undergo.	High
Position Pinning	on/off	Pins the position to a single point when the receiver thinks it is not moving	Off
Altitude hold	on/off	The altitude will be held if only three satellites are available so that a position can still be calculated, instead of not calculating a position.	Off
Ground Track Smoothing	on/off	The ground track is smoothed to eliminate sharp jumps.	Off
Measurement Quality	on/off	Controls whether lower quality satellite measurements are used.	On

The GPS receiver can be set to output many different packets, but only four are useful to this application. Table 5 lists the packets. Packet 1003 is updated every thirty seconds, but is output every ten seconds to reduce the startup latency before the information is known.

Table 5: Receiver Packet Information

Packet #	Output Rate	Description
1000	1 sec	Navigation message
1002	1 sec	Satellite quality and use information
1003	10 sec	Satellite azimuth and elevation
1102	1 sec	Raw measurement and satellite data

Packet 1000 is a basic navigation packet that has the receiver's computed position, velocity, navigation info, and UTC time and date information. This packet is used to display the receiver calculated information. Packet 1002 provides quality information on each satellite and indicates whether the receiver used the satellite in its own position calculation. This packet is used to determine exactly which satellites the receiver used for comparison with the algorithm developed for this thesis. This packet is not needed in general, and can be dropped if the satellite information is not utilized. Packet 1003 provides azimuth and elevation information to all satellites above the horizon. This information is used by the tropospheric correction algorithm. Packet 1102 provides raw measurement information and satellite data words for all satellites currently tracked. This packet contains the pseudoranges that are used to calculate the receiver's position. The full packet formats are described in appendix B.

3.6 Antennas

Two commercial antennas have been used with the receiver during testing, both made by Micropulse. The model 12700 is a robust antenna that was designed for airplanes. The antenna can be securely mounted using four bolts and a ground plane is built into the antenna for improved performance. The model 12700 has a built in 26 dB ±3 dB amplifier supplied by +5 V provided on the antenna cable. The model 12700 is fairly large (3.4x2.2x0.65 inches) and heavy (114 g), The connector is a through hole TNC that comes straight out the bottom of the antenna. The connector adds another 0.6 inches to the depth of the antenna, making the total depth 1.25 inches. The size and weight make the model 12700 difficult to use in smaller rockets. This antenna works great as the ground station antenna, where its disadvantages do not offset its great performance.

The model 13800 is small magnetic mount antenna designed for cars and other similar applications. It was bought specifically for small amateur rocket launches. The model 13800 is 2.15x1.61x0.66 inches, and weighs 52 g. A short connector cable comes off the side and is terminated with a SMA connector. A 26 dB +5 V amplifier is built into the antenna. The model 13800 does not have a ground plane which reduces its performance slightly when it is not mounted on a metal surface.

The antenna that will be used for ASRP sounding rockets is a wrap around patch antenna. This will provide continuous signal coverage during any rocket dynamics. This antenna has not been fabricated as of the writing of this thesis, and so testing and evaluation has not been possible.

Chapter 4: Software

4.1 PC Program for Data Archiving and Calculation

The software is a MS-DOS® program that was written to archive and process the data output by the receiver. The executable is called *gpsdis.exe*. The program has four different modes to handle different processing needs. The mode is set when the program is started using command line options. The modes are *'disk'*, *'real'*, *'archive'*, and *'fix'*. Figure 8 shows a flowchart of the program.

'Disk' mode is used to process information that has been previously saved. This

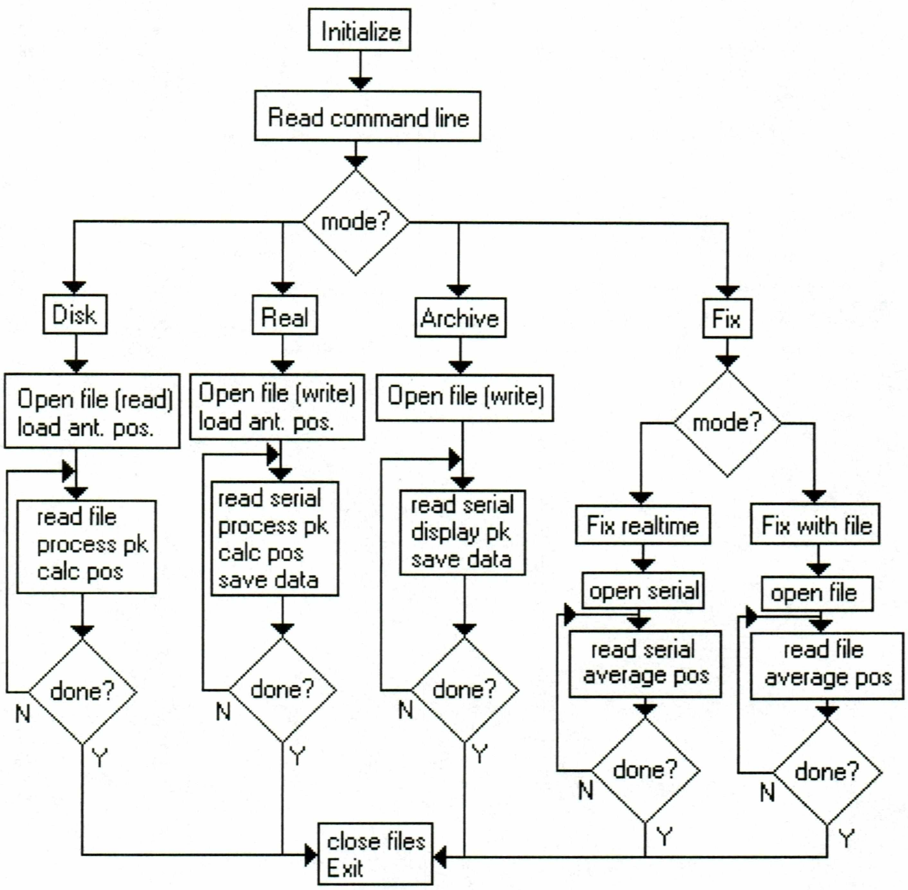

Figure 8: Program Flowchart

mode reads all the information from data files. *'Real'* mode is used to process information in real-time, taking the information from the serial ports. *'Archive'* mode saves the serial data on the hard drive for later use. No processing is done except to display information on the screen. *'Fix'* mode is used to find the antenna location. This mode can use a data file or run directly from the serial port.

The program has multiple options that range from slight tweaking to using an entirely different algorithm in the position calculation. The options are provided to maximize the quality of the data output in a wide range of situations. The program design included the ability to do non-differential calculations, so if ground station data is unavailable for some reason a rough position can still be determined. The processed data from the program is output into a tab delimited text file. This file can then be read into Matlab, Excel, or any other program for plotting and analysis. See section 4.1.6 for the details of the file format.

The software is designed to process and display both real time and saved GPS data. The program is written in C++ and designed to run in MS-DOS®. It will run in a MS-DOS® box but MS-Windows® can pause it long enough to cause the serial buffer to overflow, so it is not recommended while saving critical data. The following is a comprehensive description of how the program operates and the algorithm used for the position calculation. The different startup modes are outlined in section 4.1.1. Hot keys that control the program are described in section 4.1.2. Section 4.1.3 describes the packets and how they are processed. The position calculation is examined and the position algorithms and filter are explained in section 4.1.4. Finally, the program's display and the output file formats are described in detail in sections 4.1.5 and 4.1.6.

4.1.1 Startup

For versatility, the program has four different modes that are invoked with command line arguments. If just the mode word is used, defaults are used for the rest of the command string. For example, '*gpsdis -archive*' would result in '*Archive*' mode being started with two serial ports and generic file names. Available options are shown in Table 6 and explained in each mode section below. When the program first starts, the pause flag is set to prevent the program from processing any packets. This allows any changes in the program configuration to be made before processing starts.

'Fix' mode is used to find the location of the ground station antenna. This mode can utilize input from a saved file or real time data on serial port one. The position information is averaged over a maximum of 18 hours and then the final position is saved to disk when the program exits. The position is saved in both the text mode data file for visual inspection and in the binary '*ant_pos.gps*' file for input during a '*Real*' or '*Disk*' mode run. The position information used is the receiver calculation, not the software calculation. If only the command word is used without the input file name, '*Fix*' mode will run from the serial port. If a file name is given, '*Fix*' mode will process the file.

Table 6: List of command line arguments

-fix ['*input.file*']
-archive [1/2] ['*outfileG1*' ['*outfileG2*']]
-disk '*infileG1*' '*infileG2*' '*outfileD*' [[-e/-de] '*fileE1*' '*fileE2*']
-real '*outfileG1*' *outfileG2*' '*outfileD*' '*outfileE1*' '*outfileE2*'

'*Archive*' mode is designed to save GPS data from the serial ports for later use. Only good GPS packets are saved. Both the header and data checksum are checked before it qualifies as a good packet. If either checksum is wrong, the packet is dropped and not saved. No processing is done on the data except that needed to display useful information from the packets on the screen. '*Archive*' mode can be run with one or two serial ports. If run in single port mode it uses serial port one. The default mode activates both serial ports. The default file names are *'raw1.gps'* and *'raw2.gps'*. To run in single serial mode add a '1' after '*-archive*'. To change the default file names, put them after the port number. For two port mode, add a '2' after '*-archive*' and two file names must be given or an error message will be generated.

'*Real*' mode takes data from both serial ports and processes it in real time. Serial port one is the ground station GPS and serial port two is the flight system GPS. The incoming data is scanned for GPS packets, and each good packet is put into its respective message processing queue, ground or flight. By default the data is saved to the disk as it leaves the processing queue. This can be turned on and off if desired. After the data is queued up, the program looks at the time stamp on the packet at the front of each queue. Each GPS receiver outputs the packet sequence once a second, but they may not be synchronized. If the packets are synchronized, i.e. the difference between the two is less than one-half second, the messages are sent to the position calculation module. If they are out of sync, the old packet is dropped from the queue and the program loops. There are two options in starting this mode. The first is by just using the mode string and using the default file names. The second is to define all the file names. The default file names are '*raw1.gps*', '*raw2,gps*', '*position.gps*', '*raw1.eph*', and '*raw2.eph*' respectively.

'*Disk*' mode is the same as real mode except that the data is read from the disk instead of the serial ports. Saving to file is disabled in '*Disk*' mode. One extra option available in disk mode is the loading of ephemeris data from the disk as well. This allows position processing to begin at the start of the file instead of waiting until the information can be stripped from the GPS data packets. It is recommended that files be processed twice, the first time to strip out the ephemeris information and the second time using the saved ephemeris information. '*Disk*' mode can be started four ways. The first is to use just the mode command, default file names are used and ephemeris information is not read from the disk. The default file names are the same as for real mode. The second is to define the main data input and output file names. The default names are used for the ephemeris data and they are not read from the disk. The third is to define all the file names and use the '-e' option. Ephemeris data is not read from the disk. The fourth option is to define all file names and use the '-de' option. In this case ephemeris data will be read from the disk on startup.

4.1.2 Control

Once the program is running there are many things that are controlled using keyboard shortcuts. Table 7 lists active keys with a brief description of their use. The commands can be put into two groups, those that result in an action in the program and those that prompt for user input.

All the keys that prompt for user input gather information that affect the position calculation. The acceptable range and the units are given for all input values. The rest of the active keys have two general functions, toggling a control flag to modify program

Table 7: List of hot keys and their function

Key	Action
q	Quits the Program
s	Toggles saving on and off
d	Toggles between displaying ground and flight data
c	Clears the error/message line
e	Saves the current ephemeris data to the hard drive
F5	Toggles between normal and differential mode
F6	Toggles between outputting Receiver, Bancroft, and Linear data to the data file
F8	Toggles receiver stats use on and off
F9	Toggles Kalman filter on and off
p	Pauses the program
f	Toggles queue flushing on and off
> <	increase and decrease the delay time in disk mode
l	set the limits used in calculation module
a	enter the antenna position

t	set the parameters for the tropospheric correction

function, and carrying out a direct action. All of the toggling keys have an associated flag displayed on screen to indicate the current state.

Saving is turned on and off for both data streams at the same time. The same receiver data set that is being displayed will be output into the data file if receiver data is chosen. The ephemeris and correction data are saved to the hard drive upon program exit, so usually it is unnecessary to save it manually. This is a hold-over from an earlier version of the program and was left in case it might be needed at some point.

4.1.3 Packet Processing

GPS data that is received by the serial port is processed and placed into the data input queue. Figure 9 is a flowchart that shows how the serial data is transferred from the serial port into the program. An interrupt service routine responds when a byte is received by the serial port and places it into the serial buffer. The interrupt routine makes the program easier to write because the serial port does not have to be continuously polled for data. As the main program has time, it scans the serial port buffer for data packets. When a complete packet is found the header and data checksums are checked. If they are good the packet is placed in the queue. If the main program takes too long to process the serial port buffer, the buffer will overflow and data will be lost. This will not happen if the program is running in MS-DOS® with an Intel® Pentium II® 300 or faster because there is plenty of processing power to finish all the calculations in time to receive the next packet. All buffer overruns that occurred during testing were the result of MS Windows® interrupting the MS-DOS® box for too long to multitask.

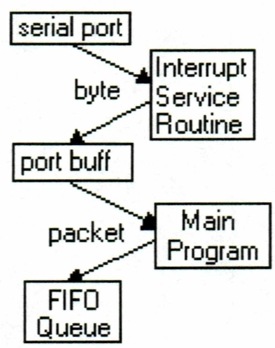

Figure 9: Flowchart of Serial Input Path

Data files that are read from the hard drive are scanned one packet at a time and placed directly into the queue. The checksums are not rechecked and the data is assumed to be good.

As the packets are placed in the queue, either from the hard drive or the serial port, they are processed by the program. Figure 10 is a flowchart showing what happens to each packet. All the data in a packet is stored in signed and unsigned integers. When the packet is processed the information is extracted, scaled, and converted to the proper type. After a packet is processed the updated information is displayed on the screen.

Packets 1000, 1002, and 1003 all contain supporting information, and so they are all processed as soon as they are received. Packet 1102 is the main data packet with all the raw data in it, so it gets special consideration. First the useful supporting information is taken out of the packet, then it is passed on to the satellite data word processing module. The two data words are the part of the 50 bps telemetry data stream from the satellite that was received since the last packet. Not every packet has data words, since

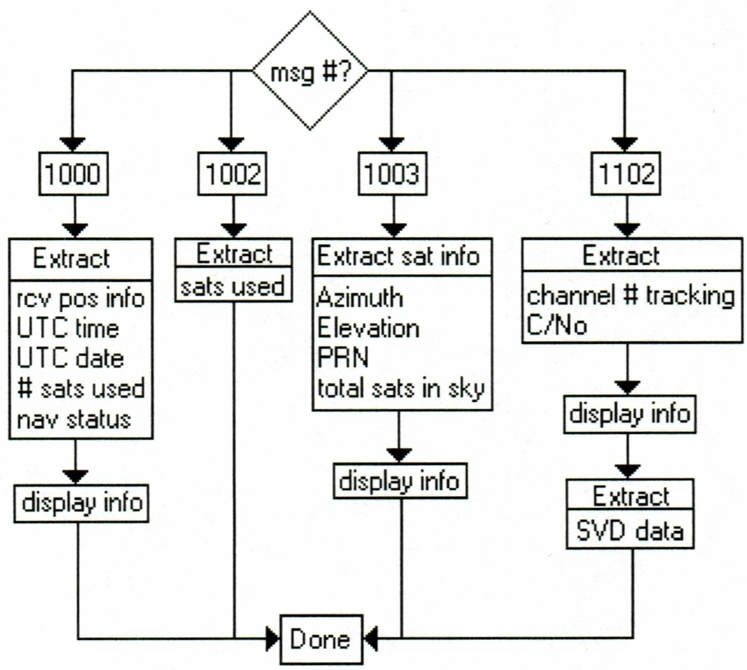

Figure 10: Flowchart of Packet Processing

the packet has more data output capability than is used by the telemetry stream. The data word processing module extracts the ephemeris and other information from the data words as they come in. Packet 1102 is finally held, to be used by the position calculation module. The other packets are saved to the disk if saving is on and removed from the queue after processing.

4.1.4 Position Processing Algorithm

All internal calculations are done in SI mks units. The position is calculated in the Earth Centered, Earth Fixed (ECEF) coordinate system. The coordinate system is fixed to the earth so that it rotates as the earth rotates. This makes a position (x,y,z) always have the same location on the earth. After the position is calculated it is converted to geographic coordinates (Lat, Lon, Alt) for display and some secondary calculations. All angles are output in degrees, distance measurements in feet, and speed in miles per hour.

The position calculation module takes two synchronized data packets and calculates a normal or differential position. See Figure 11 for a flowchart of the process. The time stamp from each packet is extracted first. Satellite selection is next and is based on the limits that have been set by the user. The limits used are satellite

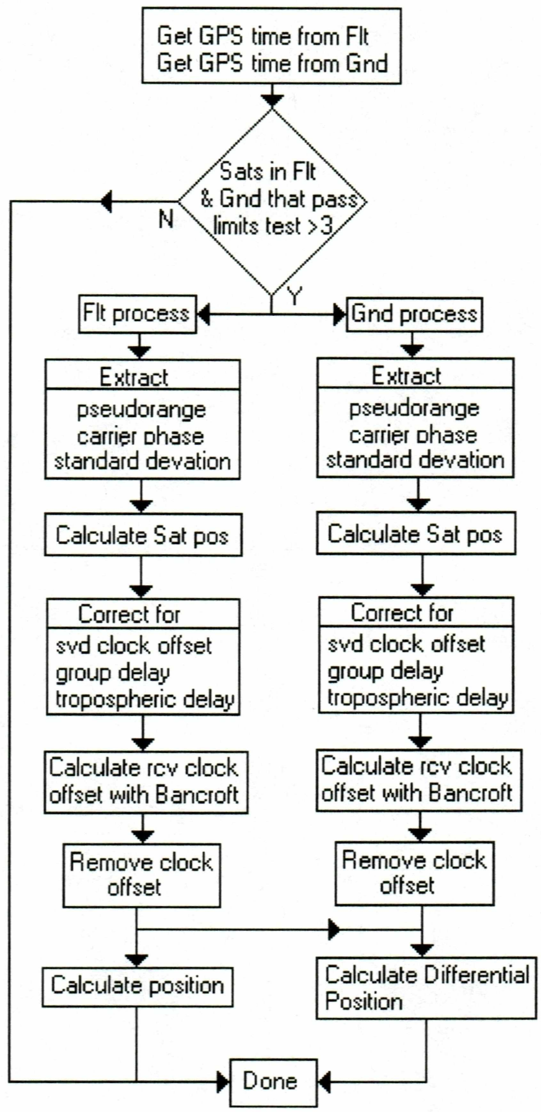

Figure 11: Flowchart of Position Calculation

elevation, receiver status indication, measurement C/No, and code phase standard deviation. All of the limits can be set by the user. The status indicators tell whether the receiver thinks that the measurement was good. Several different flags are lumped together in this category and some are more serious than others. For this reason, removing the status requirement and relying on the other limits may improve the position calculation because a measurement may still be usable when the flag is set.

All satellites that pass the limits are then used in the position calculation. If the module is in differential mode, then the same satellite must pass all the limits in both data sets. For each good satellite, the pseudorange, carrier phase, and pseudorange standard deviation are extracted from the packet for use.

After the data set from the usable satellites is gathered, the information is processed to get it ready for the position calculation. The summary that follows provides the framework of the process that will be explained in greater detail later. First the time is corrected using the satellite clock correction parameters, then satellite positions are calculated. The satellite positions are then rotated to account for the rotation of the earth during the time it took for the signal to travel from the satellite to the earth. Next the pseudorange is corrected by the calculated satellite clock offset amount and a tropospheric correction factor. Then the receiver clock offsets are solved for and removed. If enabled, the pseudoranges are smoothed using the carrier phase information.

The satellite positions are calculated using the ephemeris for each satellite and the estimated transmission time of the signal. The transmission time is generated based on the pseudorange and the speed of light. The position algorithm is straightforward and can be found in many locations, one being Leick [75]. The satellite positions are rotated to account for the rotation of the earth during the signal transmission. This is done to correct the satellite positions for the movement of the ECEF coordinate system. The satellite clock offset is modeled with a third order polynomial fit and the coefficients are transmitted in the navigation message [NAVSTAR 40]. These coefficients are then used to correct the pseudorange and the GPS time stamp by the offset amount. This whole set of calculations is repeated several times for maximum accuracy as the different calculations are interrelated.

Tropospheric corrections are important for rocket flights exceeding 10 km because the flight receiver will be above the troposphere and signals received by the flight receiver will not have the tropospheric delay experienced by the ground receiver. To minimize the degradation of the positioning solution, the delay is estimated and removed. The tropospheric correction is based on an equation by Strang and Borre [493].

$$dT = 0.002277\frac{1 + 0.0026\cos(2\varphi) + 0.00028H}{\cos(z)}\left(P_0 + \left(\frac{1255}{T_0} + 0.05\right)e_0\right) \qquad (1)$$

This equation uses the latitude ϕ, zenith distance z, air pressure P_o, height H, temperature T_o, and partial pressure of water vapor e_o to calculate the delay dT. The values needed by this equation are calculated or taken from the following inputs: current user position, the elevation of the satellite, atmospheric pressure, temperature, humidity, and the height of the pressure, temperature and humidity measurements. The pressure, temperature, humidity and height of measurement can be entered by the user for maximum accuracy. The defaults are: 1 atm of pressure, 44° F, 50% humidity, and an altitude of 450 feet for the measurements. The defaults are a compromise between summer and winter in Fairbanks, Alaska. The results from this equation are good and are expected to be competitive with more complex calculations.

The receiver clock offsets are removed by using the Bancroft algorithm, which is explained later. The single set position is calculated and, as a byproduct, the clock offset is solved for as well. This offset is then subtracted from each pseudorange. The clock offset is not common between the receivers, so it needs to be removed for the differential adjustment to the Bancroft algorithm to work. This is not necessary for the Linear algorithm, because the clock offsets get combined and the combination is solved for in the process of the position calculation. The removal of the clock offset has no harmful affect on the Linear algorithm.

The carrier phase information can be used to improve the pseudoranges without solving the ambiguity [Sam]. This is based on the concept that, as long as there has been no cycle slips, the change in the carrier phase is equal to the change in the pseudorange. Since errors are present in the measurements, this will not be exactly true, and the inherent higher accuracy of the carrier phase can be used to smooth, or improve the accuracy of the pseudorange. This differential method of using the carrier phase does not require the solving of the ambiguity since only the relative change is used. The equation used is shown below. In the equation P is the pseudorange, C is the carrier phase, N is the number of data points being smoothed, subscript N is the current measurement, and N-1 is the last measurement. For the pseudorange variables in equation (2), the bold values are the smoothed pseudoranges and not raw values.

$$\boldsymbol{P}_{\mathbf{N}} = \frac{P_N}{N} + \left(\boldsymbol{P}_{\mathbf{N-1}} + C_N - C_{N-1}\right)\left(\frac{N-1}{N}\right) \tag{2}$$

In principle, the more data points used in the smoothing process the closer the accuracy is to that of the carrier phase. In practice this is not the case because of the ionosphere and cycle slips. The ionosphere advances the carrier phase and delays the pseudorange, so the change will have some amount of divergence. Large cycle slips are detected by comparing the difference of the pseudorange and the carrier phase changes to a set limit. If the limit is exceeded, a cycle slip is declared and the number of data points being smoothed over is reset to one. If a cycle slip causes a smaller change than the slip threshold, it will go undetected. The slip threshold can be set by the user and the default is 15 meters. With high quality data this limit can be reduced down to a meter. The number of data points over which the smoothing takes place is limited to prevent undetected cycle slips from continuously affecting the data, and to limit the ionosphere divergence error. The higher the value the longer undetected slips will affect the position. Reasonable values for the smoothing window are 10-100.

The satellite positions and the corrected pseudorange information are then used to calculate the receiver position. Two algorithms are used. First is the Bancroft algorithm which directly solves for the position [Strang & Borre 500]. The second is a linear differential algorithm which uses a linear approximation to solve for the receiver position [Teunissen & Kleusberg 203].

In all of the equations in this section, a superscript refers to a satellite, a subscript refers to a receiver, and no subscript or superscript on a position variable refers to the current receiver position. The Bancroft algorithm takes the basic pseudorange equation (3) and squares both sides. The pseudorange is P, (x,y,z) is the ECEF position, and '*cdt*' is the speed of light times the receiver clock offset. When rearranged as in equation (4), it

$$P^k = \sqrt{\left(X^k - X\right)^2 + \left(Y^k - Y\right)^2 + \left(Z^k - Z\right)^2} + cdt \tag{3}$$

$$\begin{aligned}
&\tfrac{1}{2}\left(X^{k^2} + Y^{k^2} + Z^{k^2} - P^{k^2}\right) - \\
&\left(X^{k^2} X + Y^{k^2} Y + Z^{k^2} Z - P^k (cdt)\right) + \\
&\tfrac{1}{2}\left(X^2 + Y^2 + Z^2 - (cdt)^2\right) = 0
\end{aligned} \tag{4}$$

becomes apparent that the equation can be represented with three Lorentz inner products as shown in equation (5). A position vector is represented by ***r***. Each measurement generates one of the equations, so four or more measurements can be stacked together and solved for ***r*** and *cdt*, shown in equation (6). Matrix '*B*' is the satellite positions and pseudoranges. Alpha and lambda are the Lorentz inner products of the satellite vector and the solution vector respectively. The equation is recursive because the solution vector (***r***, *cdt*) is also in Λ. Using the recursion as the basis for substitution and rearranging, a

$$\tfrac{1}{2}\left\langle\begin{bmatrix}\mathbf{r}^k\\P^k\end{bmatrix},\begin{bmatrix}\mathbf{r}^k\\P^k\end{bmatrix}\right\rangle-\left\langle\begin{bmatrix}\mathbf{r}^k\\P^k\end{bmatrix},\begin{bmatrix}\mathbf{r}\\cdt\end{bmatrix}\right\rangle+\tfrac{1}{2}\left\langle\begin{bmatrix}\mathbf{r}\\cdt\end{bmatrix},\begin{bmatrix}\mathbf{r}\\cdt\end{bmatrix}\right\rangle=0 \tag{5}$$

$$B=\begin{bmatrix}\mathbf{r}^1 & P^1\\ \vdots & \vdots\end{bmatrix}\quad M=\mathbf{I}\quad \alpha_k=\tfrac{1}{2}\left\langle\begin{bmatrix}\mathbf{r}^k\\P^k\end{bmatrix},\begin{bmatrix}\mathbf{r}^k\\P^k\end{bmatrix}\right\rangle$$

$$\Lambda=\tfrac{1}{2}\left\langle\begin{bmatrix}\mathbf{r}\\cdt\end{bmatrix},\begin{bmatrix}\mathbf{r}\\cdt\end{bmatrix}\right\rangle\quad \mathbf{e}=\begin{bmatrix}1\\1\\ \vdots\end{bmatrix}\quad \vec{\alpha}=\begin{bmatrix}\alpha_1\\ \vdots\end{bmatrix}$$

$$\begin{bmatrix}\mathbf{r}\\cdt\end{bmatrix}=M\left(B^TB\right)^{-1}B^T(\Lambda\mathbf{e}+\vec{\alpha}) \tag{6}$$

$$B^+=\left(B^TB\right)^{-1}B^T$$
$$\left\langle B^+\mathbf{e},B^+\mathbf{e}\right\rangle\Lambda^2+2\left(\left\langle B^+\mathbf{e},B^+\vec{\alpha}\right\rangle-1\right)\Lambda+\left\langle B^+\vec{\alpha},B^+\vec{\alpha}\right\rangle=0 \tag{7}$$

quadratic in Λ is formed as shown in equation (7). The quadratic generates two roots resulting in two possible receiver positions. The correct solution is determined by calculating the range from both possible positions to each satellite and comparing it to the pseudorange. The position with the smallest cumulative error is chosen as the correct position. If the satellites used for the calculation are positioned in such a way that they are close to a plane, the two positions can be a mirror of each other resulting in both positions having a small error. In this case, one of the positions will be near the earth and the other far away, so the earth's radius is used to choose the correct position.

The Bancroft algorithm is not inherently a differential algorithm. Differential corrections are added by generating a correction for each pseudorange at the base station and then applying the corrections to the flight data. For this correction to be accurate the receiver clock offsets must be removed, because this is one error that is not common with both receivers. Any other receiver errors are unknown and uncorrected.

The linear algorithm makes a linearization of the basic equation (1) to make it easy to solve. To make the linearization over a smaller distance an estimate for the pseudorange is made using the last position and is subtracted from the measured pseudorange as shown in equation (8).

$$\Delta P_i^k = P_i^k - estP_i^k \qquad \Delta P_j^k = P_j^k - estP_j^k \tag{8}$$

A single difference about each satellite is then made using the data from both receivers. The equation is linearized using unit vectors from the receiver to the satellite.
The result is three matrixes: the differential pseudoranges ($\Delta\boldsymbol{P}$), the unit vectors ($\boldsymbol{u}$), and the differential position with the combined clock offset ($\Delta\boldsymbol{r}, c\Delta dt$) . The matrix equation

$$\Delta P_{ij}^k = \Delta P_j^k - \Delta P_i^k \qquad \vec{\mathbf{u}}_i^k = \frac{\vec{\mathbf{r}}_i^k}{\left\|\vec{\mathbf{r}}_i^k\right\|}$$

$$\Delta\mathbf{P} = \begin{bmatrix} \Delta P_{ij}^1 \\ \Delta P_{ij}^2 \\ \vdots \end{bmatrix} \qquad \mathbf{A} = \begin{bmatrix} -\vec{\mathbf{u}}_i^1 & 1 \\ -\vec{\mathbf{u}}_i^2 & 1 \\ \vdots & \vdots \end{bmatrix} \qquad \mathrm{Pos} = \begin{bmatrix} \Delta\vec{\mathbf{r}}_{ij} & c\Delta dt_{ij} \end{bmatrix}$$

$$\Delta\mathbf{P} = \mathbf{A} * \mathrm{Pos} \Rightarrow \mathrm{Pos} = \left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T\Delta\mathbf{P} \tag{9}$$

can be solved in a straightforward manner for the differential position with four or more satellites as shown in equation (9). The differential position is added to the base station position to get the position of the rover. For maximum accuracy the calculation is iterated several times using the updated position to generate new estimates.

After solving for the position, the result can be Kalman filtered [Strang & Borre 543]. The Kalman filter is a method of filtering using all the available information that has been collected up to the current point, without having to re-access the data that has already been processed. The filter matrices contain all the information that is needed from the previous data. The filtering is done by predicting the next position, and then correcting the prediction based on any new information that is available. Equation 10 is the basic Kalman formula.

$$\hat{\mathbf{r}}_{new} = \hat{\mathbf{r}}_{old} + K\left(\mathbf{b}_{new} - A_{new}\hat{\mathbf{r}}_{old}\right) \tag{10}$$

Where the Kalman gain matrix is K, the observation equation matrix is A_{new}, b_{new} is the new position update, and $\boldsymbol{r}$ is the filtered position vector. In equation (10) the predicted position is subtracted from the new position that was just calculated. The difference is then multiplied by a gain factor and added to the predicted position to make the new filtered position.

Estimated variances are taken into account in the Kalman gain matrix so that higher quality information is trusted more. The variance of the predicted position is set by the user during normal operation. The variance increases from the set value until an upper bound is reached if the receiver loses lock, reflecting the loss of updating information. So the longer there has been no updating data, the more it will trust any new position information instead of the predicted position. The variance information for the data from the GPS receiver is based on the standard deviation that it reports for the individual measurements. Since the standard deviation numbers that the receiver reports are large, on the order of 50 to 200 meters for good data and in the thousands for poor data, they are used directly as the variance number just to keep the numbers smaller. The default variance that is used for the predicted position is 100 meters, but this can be changed to any value by the user.

Two Kalman filters are lumped together under the single Kalman filter option. Turning the Kalman flag on and off affects both filters at the same time. The first filter is a point position filter and the second is a full position filter.

The point position filter adjusts the raw position calculation based on the variance of each measurement that was used in the position calculation. This essentially weights the position calculation to trust higher quality measurements more. This filter carries no information from previous calculations. Each position is adjusted independently. It is still a Kalman filter but instead of operating over time like the other filter it operates over the satellite list that was used to calculate the position.

The full position filter is the time-based Kalman filter. The filter uses the position that has been calculated as the new information in the prediction and updating process. The average standard deviation of all measurements used in the position calculation are used as the variance input for the new position estimate. The variance that is used for the predicted position is set by the user. The default is 100 meters. If no new data is available, the predicted position variance is doubled each calculation until it is capped at 100,000. The predicted position is based on the speed, heading and climb rate of the last known position.

If no new data is input, the rising variance will cause the filter to slowly decay the speed and climb rate to zero. This limits how far the filter will predict movement to prevent it from going too far during a longer dropout, but will allow it to coast over small dropouts. If the flight receiver trajectory turns or accelerates sharply during the dropout, then the prediction will have more error. If this a problem, the multiplication factor can be changed so that the variance climbs faster when there is no new information, or during processing the program can be paused and the variance set higher for just that section.

After the position is calculated, the speed, heading, and rate of climb are calculated using the last two data points. If the Kalman filter is enabled they are filtered along with the position.

4.1.5 Display information

Figure 12 is a screen shot of the program running in disk mode. In this figure the normally black background has been turned white and the normally white text has been turned black.

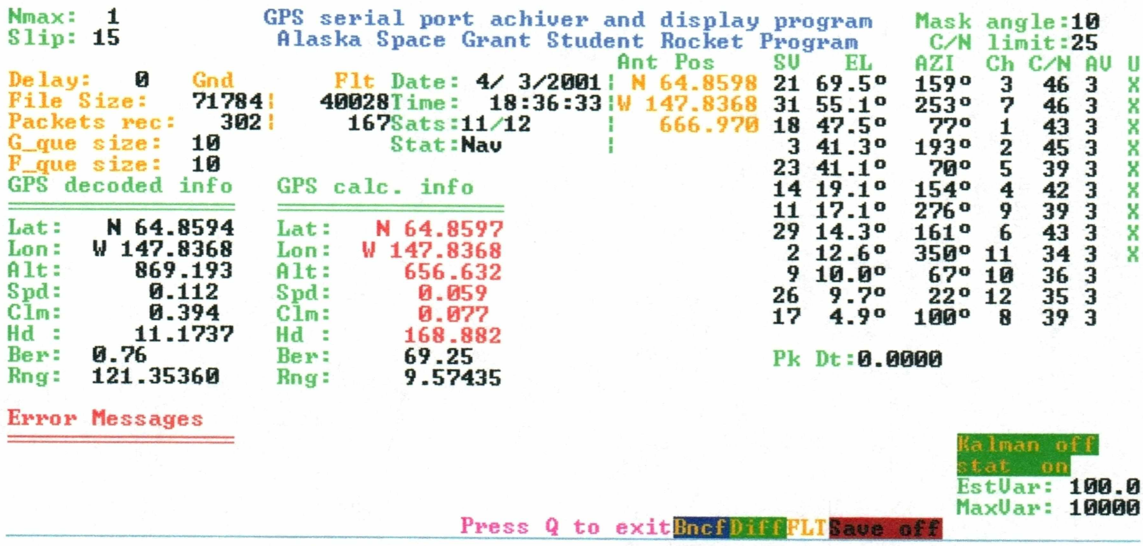

Figure 12: Screen Shot of the GPS Program

The right of the screen displays the satellite information. It lists the satellite number, elevation, azimuth, GPS receiver hardware channel tracking the satellite, C/No, correction code, and whether or not the calculation module used the satellite in the last calculation. The satellites are listed in order of elevation angle, so the highest satellite is listed first. The correction code is zero if no data is available, 1 if only ephemeris data is available, 2 if only clock and ionosphere corrections are available, and 3 if all are available for the satellite. The correction data only requires one data frame to collect so it would be very rare to only have ephemeris data available.

The upper left of the display is the system area. It is indicated here if any Com ports are open. G_que and F_que list the current size of the two message processing queues. They are listed because packets are only processed synchronously, so if there is a break in one of the data streams, one of the queues will expand until all accessible memory is used up. If the program is unable to allocate more memory it will exit. To prevent this from happening during a mission, the ability to flush the queue is available. No data is lost if this is done because all is written to the disk before deletion from memory. The stale data will be flushed automatically when the lost data stream picks up again and the program resynchronizes the two queues to the same time value. The queues can grow to a combined value around a thousand before the end of memory is imminent. This is enough for over 15 minutes, so manually flushing for this cause should be rare unless there has been a major break in communication. The other use for flushing the queue is while processing files. If it is desired to process to the end of the ground station file, (e.g., if the ephemeris data is being extracted for use) then flushing will need to be turned on or the program will stop at the end of the flight file.

Packets Received lists the number of packets that have been put in the message queues. In *'Fix'* mode this is modified to list the number of packets that have been processed and averaged together. The file size lists the size of the file in '*Archive*' and '*Real*' modes and lists the amount the file currently processed in '*Disk*' mode. If no file is open '*none*' is displayed here. In '*Disk*' mode the current delay value is displayed. This is the delay in ms that the program waits before looping back to process the next packet. A delay around 300 ms results in approximately real-time processing.

Flags at the bottom of the screen are used to display status and alert messages. Flags are denoted by having a non-black background highlighting them from the rest of the text. Five flags are normally displayed and indicate status information. One indicates what calculation method is being used, Bancroft (Bncf), Linear (Linr), or Receiver (Recv). If Recv is displayed, the program is just outputting the position information from the receiver and is not calculating a position. Another indicates whether the calculation module is using a single data set (Norm) or is in differential mode (Diff). This can only be changed for the Bancroft calculation since the other two are inherently one or the other. Another flag indicates which data set is being displayed on the screen, flight (FLT) or ground (GND). Another indicates whether or not saving is on. Two more flags indicate whether the Kalman filter is being used and if receiver stats are limiting the satellites being used by the calculation module. Another three flags are alert flags that are not normally shown. The first indicates that the queue is being flushed, and the last two indicate that the flight and ground files are empty. These flags would only appear in '*Disk*' mode when the end of the respective file is reached.

Scattered around in the corners of the display are the current limits that are being used in the calculation module. The upper left shows the Nmax and slip limits for the carrier phase smoothing algorithm. The upper right shows the mask angle and the C/No limits that are being used during satellite selection. The lower right shows the variance (EstVar) that is being normally applied to the estimated new position in the Kalman filter. Also here is the max variance (MaxVar) that is used during satellite selection.

The upper middle of the display shows the date, time, and navigation information decoded from the 1000 packet. The time is UTC time corrected to AK standard time. No adjustment is made for daylight savings time. During the end of month transition there will be a nine hour period where the day will read zero. This is because no adjustment is made for month rollovers in the date correction. Next to this information is the antenna position that is being used for bearing/range and differential calculations. If the antenna position file was overwritten or damaged it will show up as an incorrect or bogus position value here. The most common problem is '*Fix*' mode overwriting the antenna position with a less accurate or zero value. Once the antenna is fixed it is recommended that a backup copy of the *ant_pos.gps* file be kept.

The middle left of the display shows the position, altitude, speed, and rate of climb from the 1000 packet. This information will drop out during a rocket flight as the GPS receiver stops calculating a solution, but is displayed for comparison purposes during testing and evaluation. Also shown here is the bearing and range from the base station to the receiver's calculated position.

The middle of the display shows the position, altitude, speed, and rate of climb, that have been calculated by the position module. Also shown are the bearing and range from the base station to this position. This can be either a single set solution or a differential solution, the flag at the bottom of the screen will tell.

The last part of the display is the error message section on the bottom. Any error messages that do not require program shutdown are displayed here. This is also the location where user input takes place.

4.1.6 File Formats

The main data files that store the GPS data are binary files. The default file names are *raw1.gps* and *raw2.gps*. The format is [msg size][record][msg size][record].... The msg size is a 16 bit integer and corresponds with the next record in the file. This format allows a simple binary read of the data while still allowing different sized messages. Each record is one packet from the GPS receiver. The full packet is saved with the header and checksum information. Both the header and data checksums are checked when the packet is first received from the serial port and if they are bad the packet is not saved. The checksums are not rechecked when the data is read from the file. File reads are checked to make sure the full packet was read. That way a damaged file that ends in the middle of a packet will not cause bogus data to be processed.

The satellite data is split into two binary files. The first file holds any ephemeris information that has been collected on the satellites, and the second file holds clock and ionosphere correction information. The default file names are *raw1.eph* and *raw2.eph*. The ephemeris file has the following format: [prn][number of records][rec1][rec2]... [prn][number of records][rec1].... Only satellites that have data are listed in the file. The ephemerides are updated every hour, so during a long track multiple records for each satellite will be downloaded. The corrections file has the following format: [prn][rec][prn][rec].... Only one data set is saved for each satellite since the information is not updated quickly. File reads are checked for size to make sure that the file was not damaged.

The antenna position file is a binary file that has the ECEF coordinates of the antenna. This file is generated by *'Fix'* mode and must be present in *'Disk'* or *'Real'* modes since the differential calculation needs the antenna position.

The position data file is a tab delimitated text file. A header line is included that explains what each column is. The format allows the file to be easily imported into a spreadsheet for plotting and displaying. The default file name is *position.gps*. This file has two purposes, in *'Real'* and *'Disk'* modes it is the position output file, and in '*Fix*' mode it contains the fixed antenna position. In *'Real'* and *'Disk'* mode this file has one line for each position output. The values output are Latitude, Longitude, Altitude, ECEF position, Heading, Speed, Rate of Climb, Bering, Range, Satellites Used in the Calculation, and Time. The time stamp is based off of UTC time but is corrected to AK standard time.

The debug file is a text file that lists debugging information during programing and testing. The file name is *debug.gps*.

4.2 Storage Board Microcontroller Program

Several programs are used to control the 68HC11F1, the embedded microcontroller in the Storage Flight Board. The main program provides all the needed functionality for normal operations. Two more utility programs have specific uses. The board outputs a logic level serial stream so the RS-232 converter board is needed to communicate with a computer.

The main program has two operating modes: '*Save*' and '*Dump*'. The mode is chosen by setting a jumper before powering up the board. If the board is already powered the HC11 must be reset before the new mode will be activated. If memory is to be saved, the board must be switched to *'Dump*' mode before powering up. Memory is not erased, so all data will not be lost if powered up in the wrong mode, but if the GPS receiver is connected any data output by the receiver will overwrite what is currently in memory.

'Dump' mode is used to dump the data in memory to the serial port. The GPS receiver is left powered off in this mode. Dumping starts immediately so the computer needs to be ready to accept data before the board is powered on. Memory is not erased or overwritten, so this is a safe mode to have the HC11 in if the data in RAM needs to be

saved. There is no end of data flag, so all of memory is dumped. The data dump will need to be monitored at the computer to stop the download at the end of the current data set. The serial jumpers must be set so that the HC11 serial out is communicating with the PC serial in.

'*Save*' mode is used to save GPS data to RAM. In this mode, after the HC11 is initialized, the GPS receiver is powered up and any data that it outputs is saved to RAM. The serial jumpers need to be set so that the GPS serial out is connected to the HC11 serial in. The HC11 also monitors battery voltage so that it can shut down the receiver and go into standby if the voltage drops too far. The cutoff voltage depends on the battery used. It is currently set at 6.5 V, allowing the battery some headroom to run the board in standby before the voltage drops below 5 V resetting the HC11. If the HC11 resets it will try to power up and save data as if it has been normally restarted. If it is successful the start of memory will be overwritten. This should never become a problem though, since the high power draw of the receiver would drop the voltage to the cutoff point again. It takes several seconds for the GPS receiver to initialize and start sending packets. To provide a status output the program monitors the incoming data stream and looks for message packet 1000. This packet has information that indicates whether or not the receiver is locked up. When the receiver locks up a red LED is illuminated on the board. Each time a 1000 packet is processed a green LED on the board flashes to show that packets are being received by the board.

The first utility program, *erase*, is used to erase the contents of the SRAM chips. When the data is being dumped by the main program it has no idea where the current data set ends, so if a smaller data set was recorded over a larger data set there will be an abrupt transition between the two during downloading. Erase provides a way around this by explicitly erasing the old data from memory. It resets all memory locations to the default 0xFF. Erase runs from RAM, and therefore doesn't overwrite the main code that is stored in EEPROM. It is recommended that memory be erased before the board is flown.

The second utility program, *tmem*, is used during testing and debugging of a new board. The purpose of this program is to test the function of the memory chip or chips by writing a test pattern to all memory locations, and then reading the test pattern back and record the number of errors. This is used to test whether all the address lines are connected properly, and, by inserting a power cycle between the write and read, testing the battery backup. The most useful way to use the program is to use PCBug to read the status memory locations, 0x100-0x10F, during the memory read. This allows two additional pieces of information to be collected; where in the chip the problem is, and if the error count perfectly overflows. Sixteen bits are used to store the error count, so if a multiple of 65,536 memory locations are bad, it will indicate zero errors. An example of this would be if the chip was completely non-operational.

The HC11F1 microprocessor does not have Motorola's Buffalo terminal interface built in, so PCBug must be used to program the F1. PCBug comes with the HC11E9 evaluation kits or can be downloaded from Motorola's website. PCBug works by downloading a small bootloader program onto the microcontroller on startup. PCBug then uses this program to read information from the registers, control program execution, and to download programs into RAM and EEPROM. Before download, the HC11 must be put into boot mode with the boot jumper and the serial jumpers must be set to connect the HC11 and PC together. The bootloader runs using the software interrupt, so interrupt driven programs, like the main GPS code, interfere with the bootloader and can't be run at the same time. The normal bootloader is setup to run with 512 bytes of RAM. Since the F1 has 1 kbytes of RAM, a modified bootloader was created for the F1. The only change was to put the stack at the top of RAM (0x3FF) instead of in the middle of RAM (0x1FF), as it was by default. Since the source code for the bootloader is unavailable, the change was made by editing the raw code by hand with Norton Diskedit. The modified bootloader is called '*talkf.boo*'.

Chapter 5: Ground Testing

5.1 Receiver Performance

The GPS receiver outputs a calculated position along with the raw data. This information is examined first to get an idea of how the receiver performs and how the atmosphere and individual receivers affect the positioning data. The receivers have several settings that affect the internal position calculation, and the effect of the different settings were evaluated. The setup for most of the tests was a single antenna that was split to two receivers with a power splitter. This setup isolated the receiver differences between the two data sets, since that was the only difference between the two. For the last data set, two antennas were used, one for each receiver, to explore the effect of the antenna on the data.

During the testing process it was noted that the different GPS receivers always sync up and output packets at the same GPS time with the same antenna. They continue to do this most of the time with separate antennas, but one data set was recorded where they are out of sync, so it is possible for this to happen. It was noted as well that even when the receivers are using the same antenna, the same satellites are not always used for the position calculation. If the satellite is on the edge of trackability, it may not be used by both receivers. This difference is attributed to the RF section of the receivers, since if the satellite is tracked properly, and it is above the elevation mask, it should be used. The difference could come from a variation in the output of the power splitter, between the RF sections, or between the different channels in the RF section, since the same satellite is usually tracked by a different channel between the receivers.

Figures 13 and 14 show that the two receivers follow each other closely, but not exactly. Altitude differences up to 10 feet and position differences up to 30 feet occur between the receivers, compared with a position error up to 80 feet during this period. The changes that both receivers go through are the result of atmospheric and satellite errors that are common to both. The differences are the result of errors specific to the receiver as well as the result of using different satellites, as mentioned previously.

5.1.1 Dynamic Setting

The receivers have a dynamic setting that sets the expected level of movement the receiver will undergo. The dynamic level can be set to low, med, and high, which Rockwell corresponds to hiking, driving, and flying. It is expected that the setting controls how aggressively the receiver smooths and filters the raw data to generate the position information. Figures 15 through 17 show approximately15 minute data sets at each of the dynamic levels.

The graphs show that, as expected, the low dynamic setting has a much lower short term drift rate. There is one big swing in the data sample between 75 and 175 seconds that results in almost all of the range in the positioning error. This might be caused by a satellite geometry effect. Figure 17 shows that the receiver was in the process of losing and gaining a satellite during this time period. If the first minute after power up and the big swing are discounted, then the altitude stayed within a 20 ft span and the position stayed within a 10 ft^2 box.

The medium dynamic setting appears to be closer to the high setting than the low setting due to some wide swings. But if the two greatest swings are limited, then the range is a nice middle ground between the two settings. For most of the time, the position was within a 20x40 ft box and the altitude was within a 40 ft span.

The high dynamic setting wanders faster and with less smoothing that the other settings. For most of the time, the position was within a 30x60 ft box and the altitude was within a 80 ft span.

A 12 to 18 hour data set was taken at each dynamic setting to see what the long term variations were. Figures 18 to 23 show the data. The histograms show an interesting result. Over the long term, the high dynamic setting actually has a smaller 90% circle (28 ft) than the low (35 ft) and medium (41 ft) settings. This is a very interesting result, and it must arise from how the receiver is using and filtering the raw data. The time duration of the data samples was long enough that all three sets would have seen the same satellite constellation, only at a different point in the sample period.

For normal use, the short term data is more indicative of the error range that would be present. This data shows that increasing the dynamic setting increases the rms error bar around the position from about 10 ft to 50 ft. The expected advantage of the higher dynamic settings is that the receiver will follow a fast moving object better. For example if a high speed object took a sharp turn, the high dynamic setting would track through the turn more accurately than the low dynamic setting which would round off the corner with heavy smoothing.

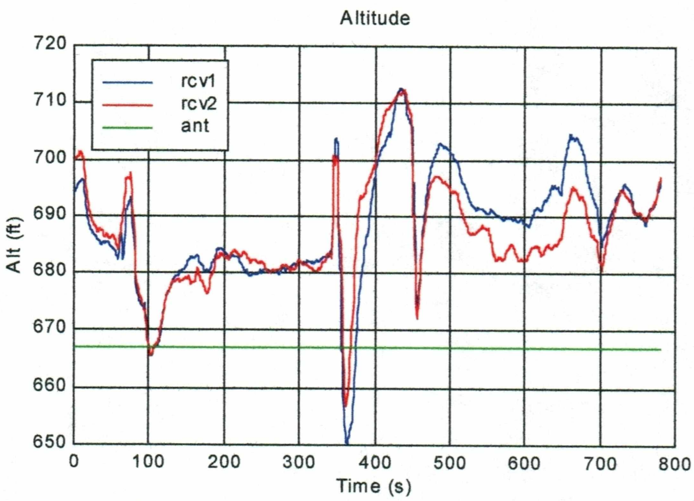

Figure 13: Altitude comparison between receivers

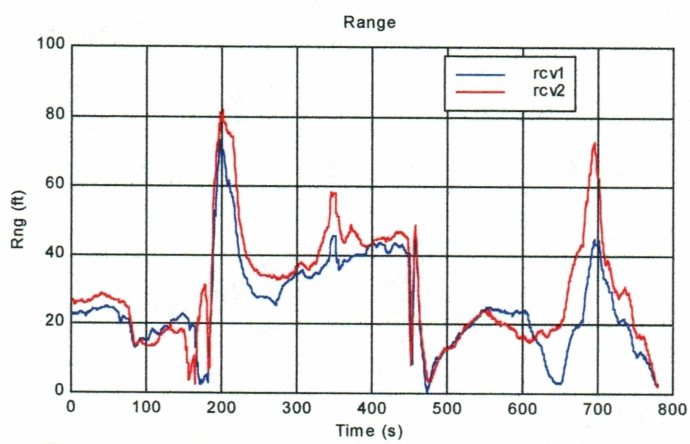

Figure 14: Range comparison between receivers

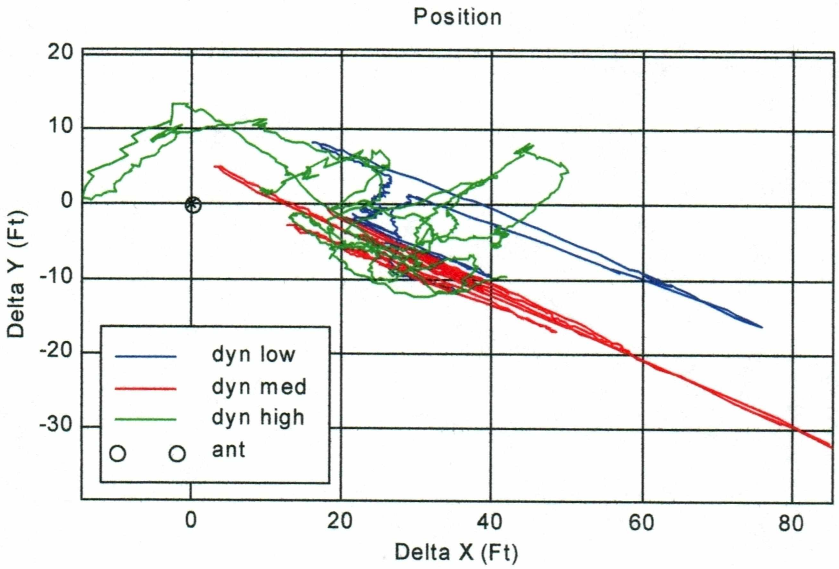

Figure 15: Position Error with Different Dynamic Settings

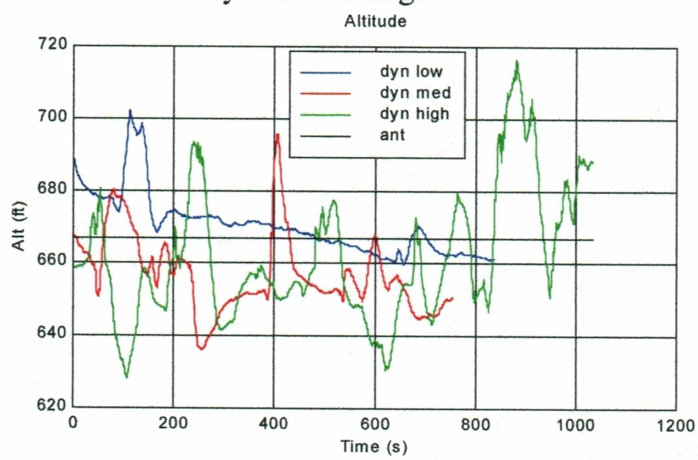

Figure 16: Altitude with Different Dynamic Settings

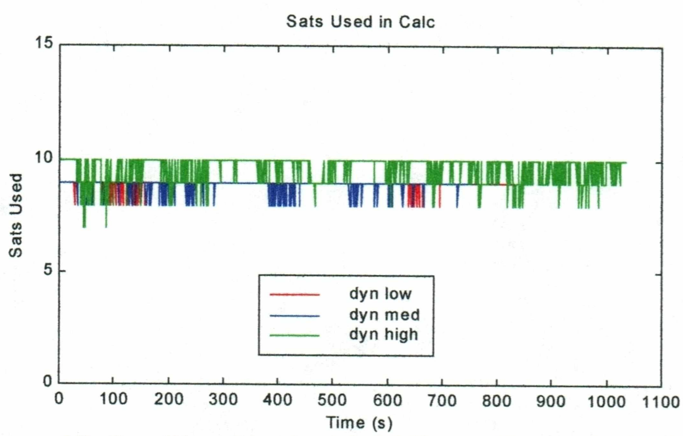

Figure 17: Satellites Used with Different Dynamic Settings

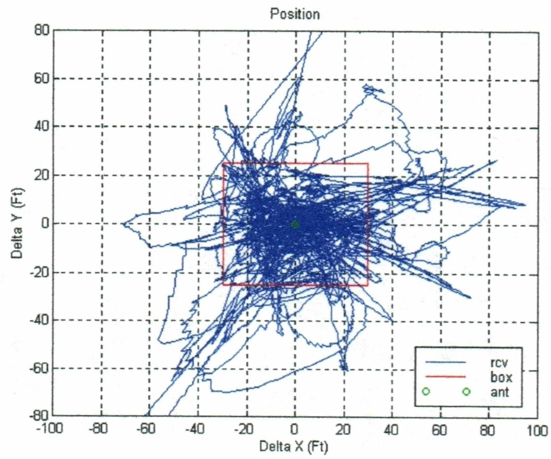

Figure 18: Long Data Set – High Dynamics

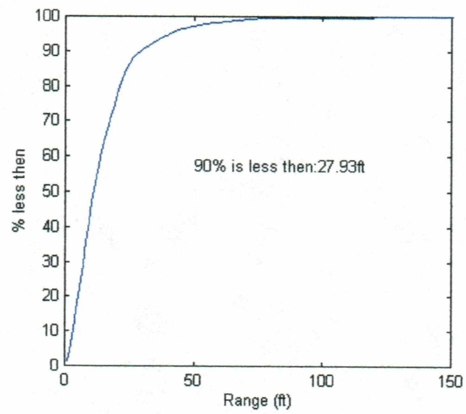

Figure 19: Histogram of High Dynamics

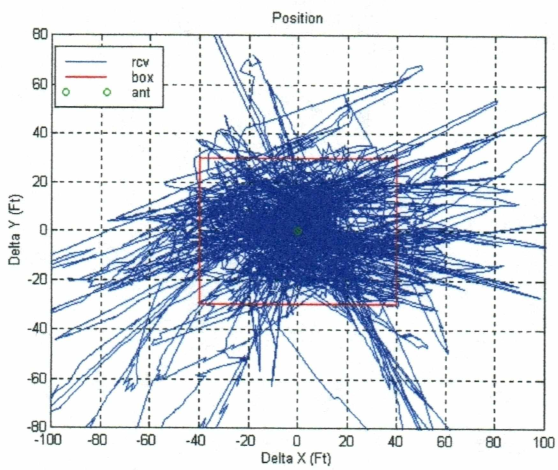

Figure 20: Long Data Set – Medium Dynamics

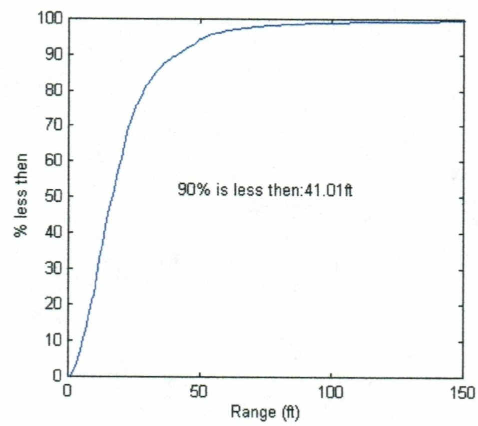

Figure 21: Histogram of Medium Dynamics

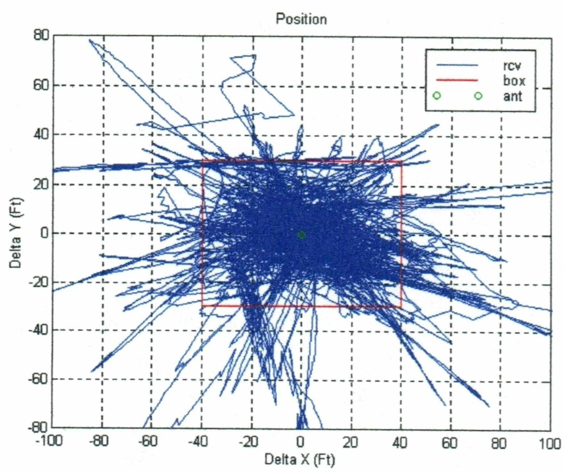

Figure 22: Long Data Set – Low Dynamics

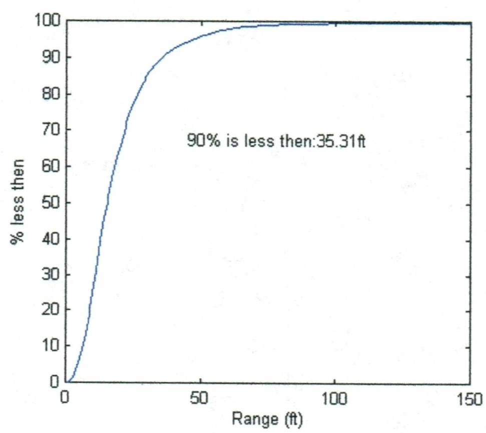

Figure 23: Histogram of Low Dynamics

5.1.2 Elevation Mask

The elevation mask limits the satellites that are used for the position calculation to those that are higher above the horizon than the mask angle. The mask angle affects the position calculation because the geometry is improved by using satellites lower to the horizon. However, the signal transmitted by low satellites travels through more atmosphere and therefore has larger errors in the measurement. This tradeoff results in a search for the optimum mask angle that results in the best positioning solution. Figures 24 through 29 show the result of collecting approximately15 minute data sets in 5° increments from 0° to 20°. The information is muddled since the satellite geometry was different for each of the data sets, but some general conclusions can be drawn.

The 0° and 5° settings have the most variation. The 5° mask has a greater variation and this is attributed to tracking less satellites even with the low mask angle. The conclusion is that these mask settings add more error from the low angle atmospheric signal propagation than is being removed by improving the geometry of the calculation. The only case where mask settings this low would be useful is if so few satellites are being tracked that lowering the mask to this setting will increase the number used in the position solution to four or five.

The 10° mask setting shows a marked improvement over the 0° and 5° settings. Some of the improvement is attributed to the increased number of satellites high in the sky during this sample period. The large number allows an improved least squares fix.

The 15° mask setting shows that higher masks are not helpful if the result lowers the number of satellites to five or less. The first half of the data set is with five satellites and the position varies rapidly over 100 ft, but once the number increases to seven, the position settles down and varies similarly to the 10° and 20° settings.

The 20° mask setting results in a position error similar to the 10° setting except that an average of seven satellites are used instead of ten. At this point satellite quality is not improving much compared to the loss of geometric information and the position is not improved over the 10° or 15° settings.

The conclusion drawn from the data sets is that a mask angle between 10° or 15° is usually best. Higher settings result in increased altitude errors. Lower settings only improve the solution if a low number of satellites are being tracked and including lower ones increases the total up to five.

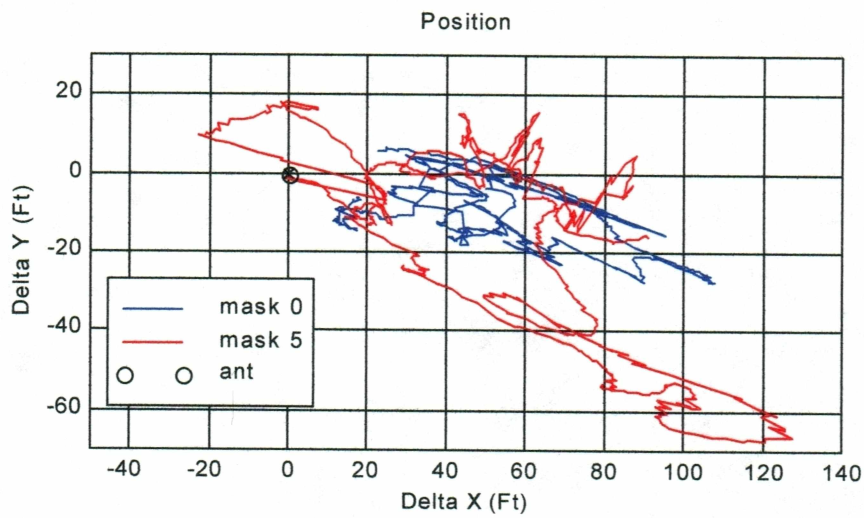

Figure 24: Position Error with 0° & 5° Elevation Masks

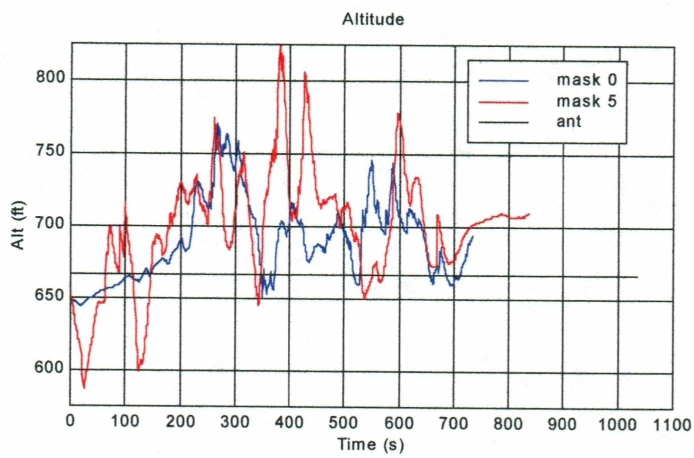

Figure 25: Altitude with 0° & 5° Elevation Masks

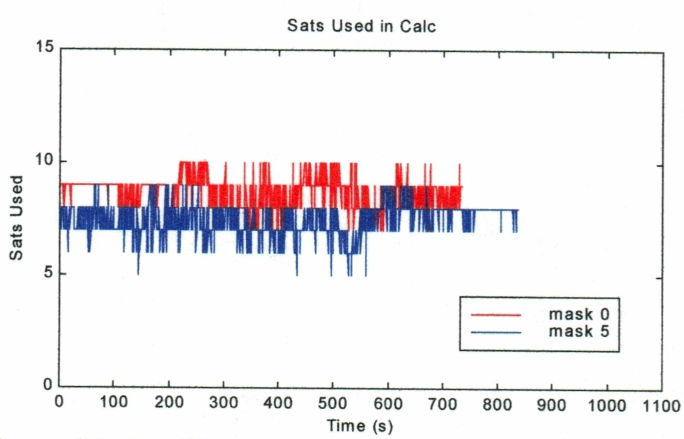

Figure 26: Satellites Used with 0° & 5° Elevation Masks

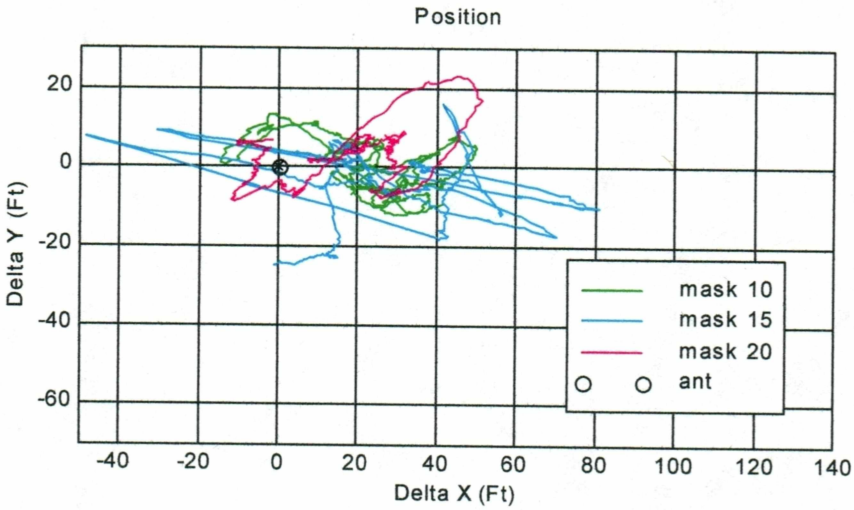

Figure 27: Position Error with 10°, 15°, & 20° Elevation Masks

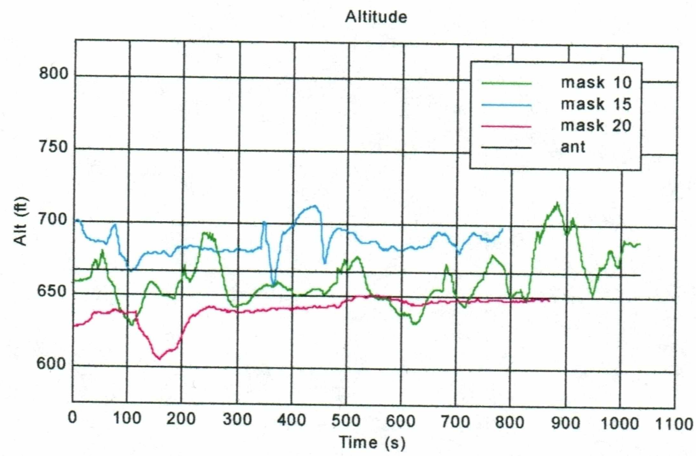

Figure 28: Altitude with 10°, 15°, & 20° Elevation Masks

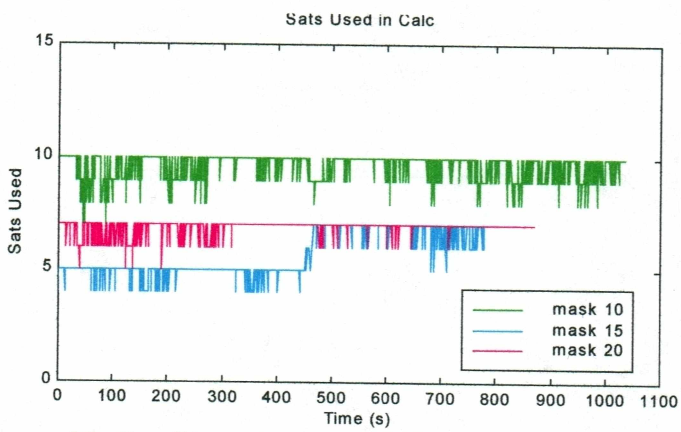

Figure 29: Satellites Used with 10°, 15°, & 20° Elevation Masks

5.1.3 Power up versus Steady State Position

The GPS receiver uses a complex Kalman filter to smooth the output position. It was expected that this filter would be sensitive to initial conditions so that even two receivers powered up at the same time with the same antenna would have some positioning difference. This difference would be expected to decrease over time as filters continue to process similar inputs.

Figures 30 through 33 show the difference between a data set collected when the receiver was first powered on and one collected after the receivers have been operating for 12 hours. The graphs show that the two receivers are tracking closer together than at first, but are still not exactly the same. This is due to the errors that are specific to each receiver. The differences may not always affect the raw input, but the Kalman filtering process will smooth the differences over a long time period such that some difference is always present. The number of satellites used was more consistent for the second data set which might have favored a closer agreement between the receivers. The data set shows that there is enough continued difference between the two receivers that they never converge to the same position.

5.1.4 Different Antennas

It would be expected that using different antennas and receivers would increase the difference between the position outputs. The result should be similar however, since the antennas are in the same locations and see the same satellites.

Figures 34 through 36 show the result of having two antennas. As was expected, the two positions do not track smoothly together as they did before, but this change was of a much larger magnitude than expected. Receiver two (red in the graph) was not tracking the satellites as well as receiver one, which may explain the increase in position error and wandering. Further testing has shown that even with two identical high quality antennas this setup produces two completely different positioning results and the data does not track together as it did with the single antenna.

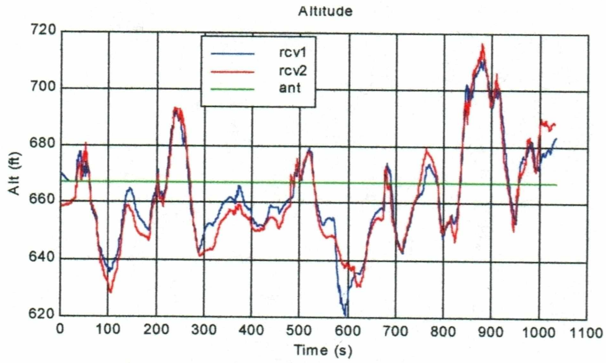

Figure 30: Altitude Power Up

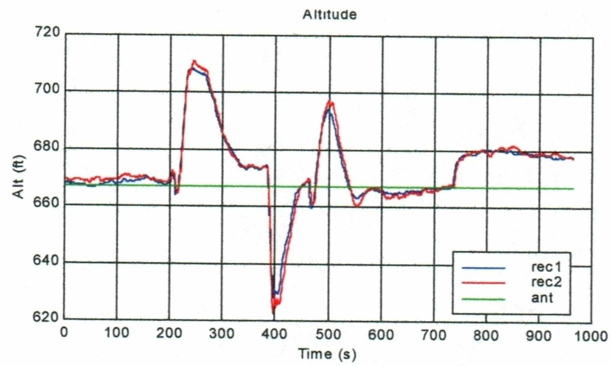

Figure 31: Altitude Steady State

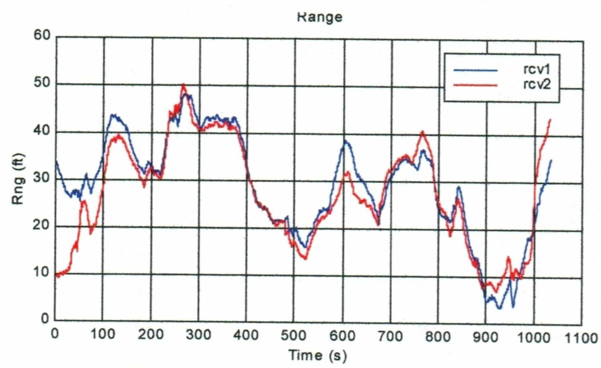

Figure 32: Range Power Up

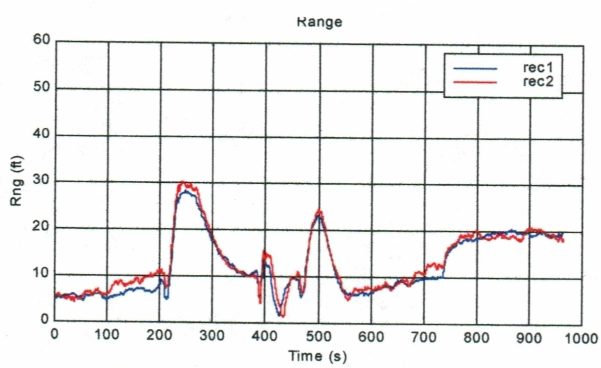

Figure 33: Range Steady State

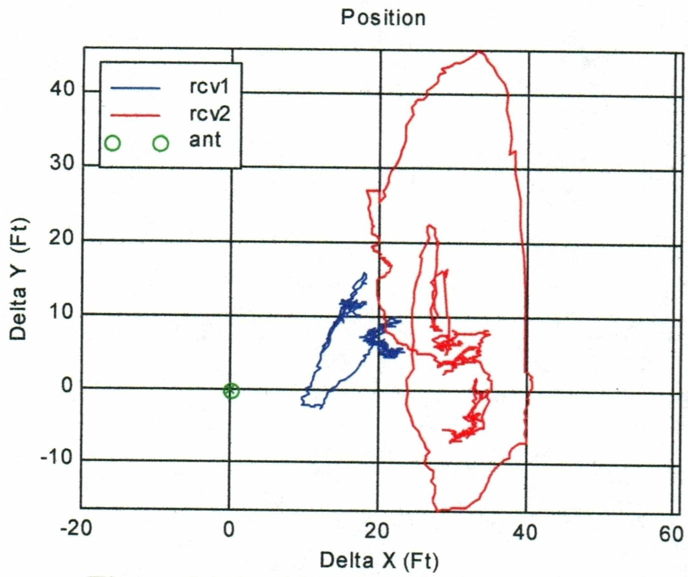

Figure 34: Position for Dual Antenna Test

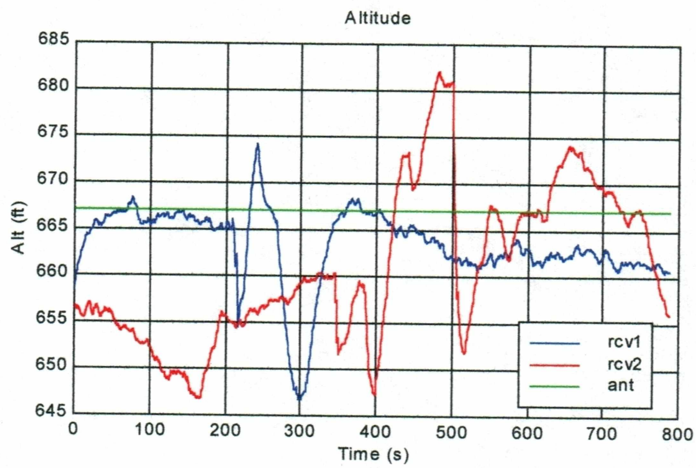

Figure 35: Altitude for Dual Antenna Test

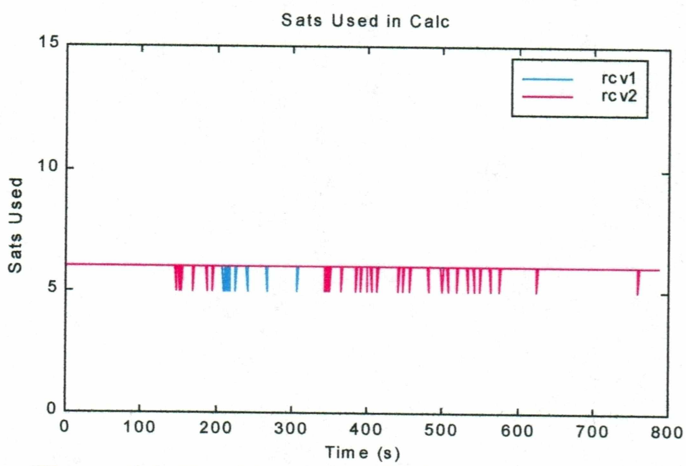

Figure 36: Satellites Used for Dual Antenna Test

5.2 Non-Differential Performance

The next step is to look at the standard non-differential calculated solution and compare it to the output from the receiver. All the calculations in this section use a 10° elevation mask and the Bancroft position calculation. To run a non-differential calculation when there is only one data set available a small trick must be used. This is because the program assumes that it will be used for a differential calculation and requires two data files on startup. The way to get around this is to use the same input file for both. A bogus file can't be used since without valid synchronous time stamps the calculation module will never be called. If differential mode is accidently used with single file input, the position output will be perfectly corrected to the antenna position. This is because the same data set is used to generate the corrections and calculate the position.

The raw calculation using all available satellites has a broad spread as shown in Figure 37. The error range can be reduced from the initial ~60 ft range to ~20 ft by dropping satellites that have large error bars on the measurement. The result is shown in the same figure. This improvement is possible because the raw calculation does not weight the measurements based on their quality, so just a few poor measurements will have a large impact on the error range.

Figure 38 compares the receiver output to the variance limited solution. The figures show that the center of the calculation is about the same as the receiver's internal calculation, but with less wandering. The solution can be further improved by adding Kalman filtering. Variance limiting has a much smaller effect with the filter enabled because the Kalman filter uses the variance information output by the receiver to weight the raw data. Figure 39 compares the filtered data to the receiver output. Filtering reduced the error range to about 10 ft.

In this data set the non-differential calculation is better than the output from the receiver. This is most likely because the receiver is trying hard to predict and smooth the movement of the antenna, so it is losing in the situation where the antenna is stationary.

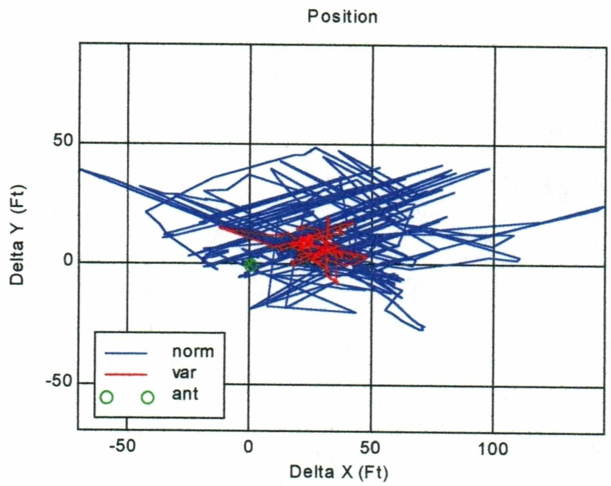

Figure 37: Non-Differential Position, Raw and Variance Limited

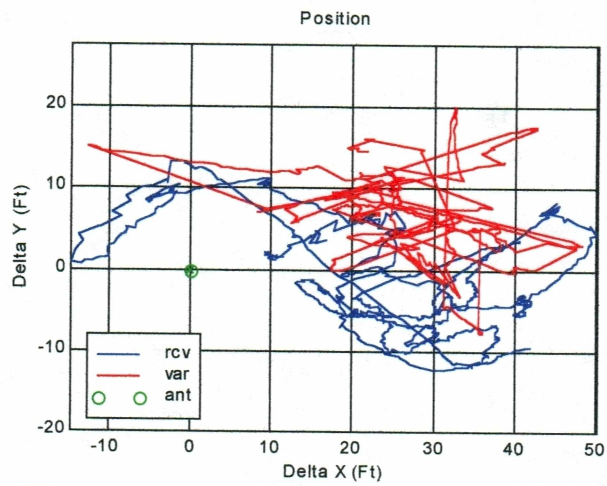

Figure 38: Non-Differential Variance Limited Position Compared to Receiver Output

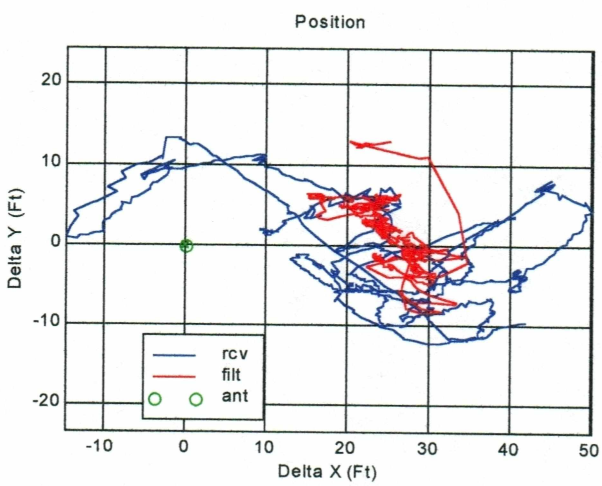

Figure 39: Non-Differential Kalman Filtered Position Compared to Receiver Output

5.3 Differential Performance

The last step is to look at the differential performance. All the calculations in this section use a 10° elevation mask.

5.3.1 Stationary Ground Test

The raw differential calculation is shown in Figure 40. The figure clearly shows how the data has recentered about the correct value. The 90% range for the raw calculation is 6.36 ft as shown in Figure 43. Figure 41 shows the improvement that is gained by adding Kalman filtering. The filtering improves the 90% range to 2.13 ft as shown in Figure 44.

The large error spikes occur when the receiver differences for a satellite diverge in such a manner that the correction actually makes the measurement worse. Figure 42 shows the result of processing the dual antenna setup. The data has been recentered as before but the error bar is much larger due to the differences in the data. The plot of the receiver outputs in Figure 34 shows how different the two data sets were, so it is reasonable that the differential calculation isn't always helped by the corrections generated from the second data set.

The raw calculated data is compared to the Kalman filtered data versus time in Figure 46. The jumps that caused by poor measurements, geometry changes, and receiver errors. The ones caused by poor measurements are easily filtered out, since those measurements are trusted very little. Some of the spikes remain, especially in the dual receiver test. These spikes are caused by satellite geometry and errors between the receivers that are not consistent.

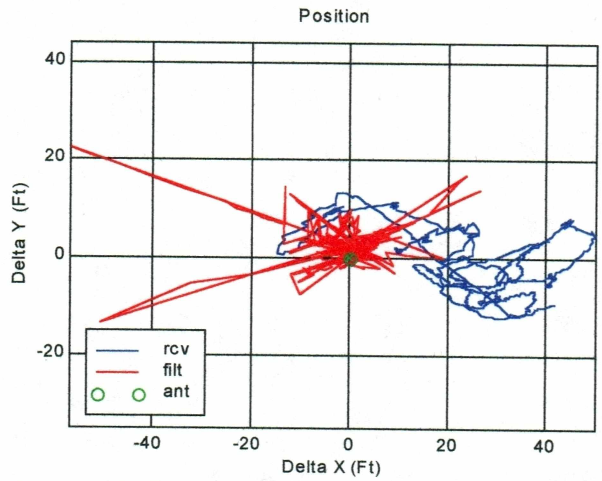

Figure 40: Raw Differential and Receiver Output

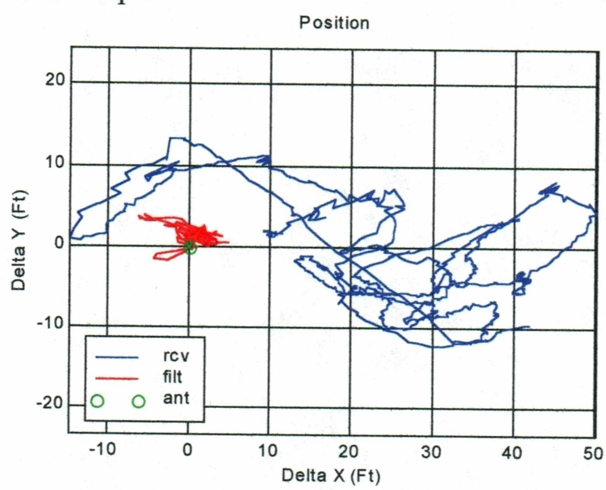

Figure 41: Kalman Filtered Differential and Receiver Output

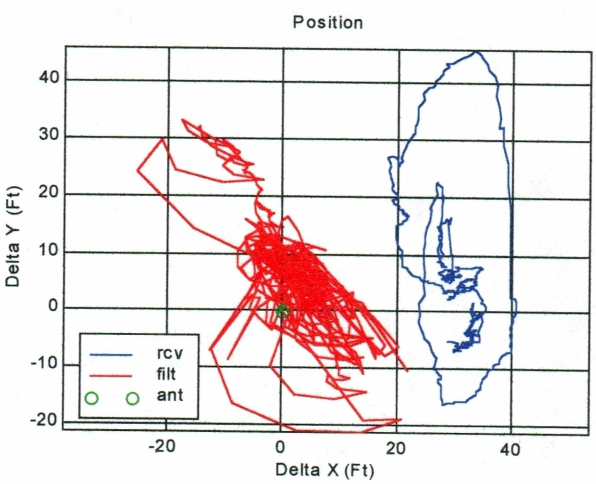

Figure 42: Kalman Filtered Differential and Receiver Output with Dual Antennas

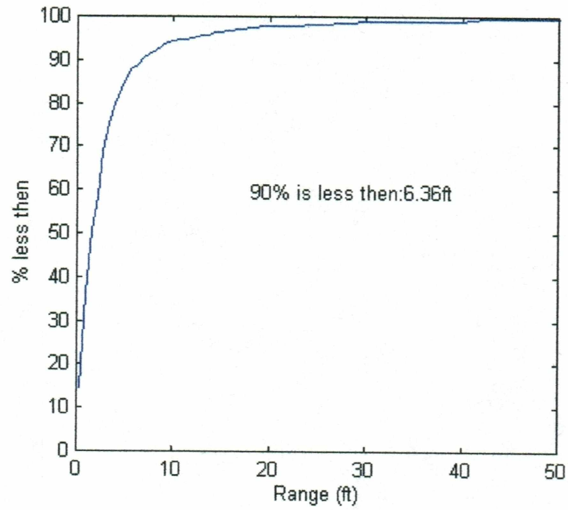

Figure 43: Histogram of Raw Differential

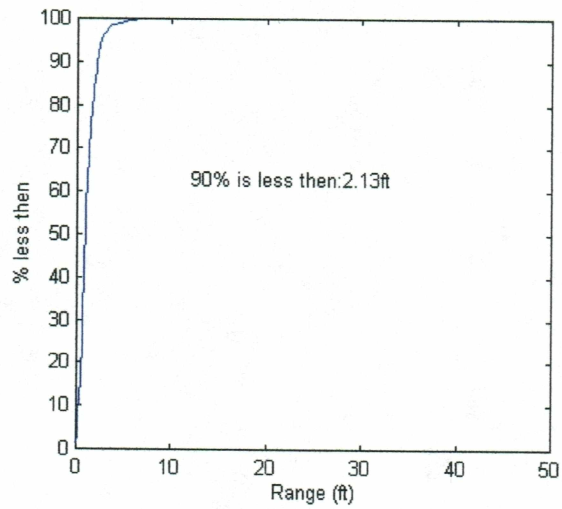

Figure 44: Histogram of Filtered Differential

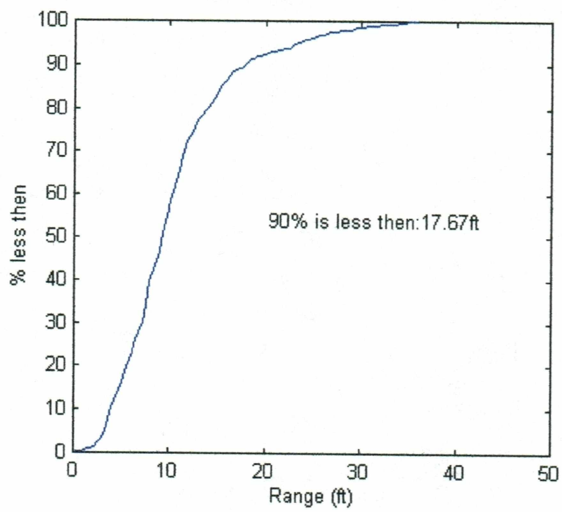

Figure 45: Histogram of Dual Antenna Differential

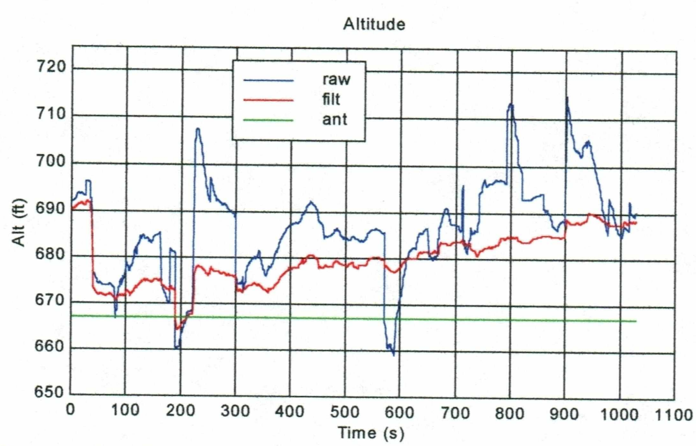

Figure 46: Comparison of Kalman filtered and Raw data

5.3.2 Moving Ground Test

This data set was taken while driving between the T-4 trailer on the UAF campus and the Old University Park School. Figure 47 compares the receiver output with the raw differential calculation. The agreement is good -- the differential change is very small at this scale. Figure 49 shows a the portion of data driving out of the parking lot and around the Natural Sciences Building. This scale is small enough to show the differential corrections. Once the receiver was moving, its advanced Kalman filter and smoothing was an advantage. The raw differential data shows small jumps that result from changing the number of satellites used in the calculation. The Kalman filter in the calculation module does not predict acceleration and rate of heading change, so the filtered data doesn't follow corners very well. For rocket flights this is not a large handicap since the nominal flight path will not have any sharp turns.

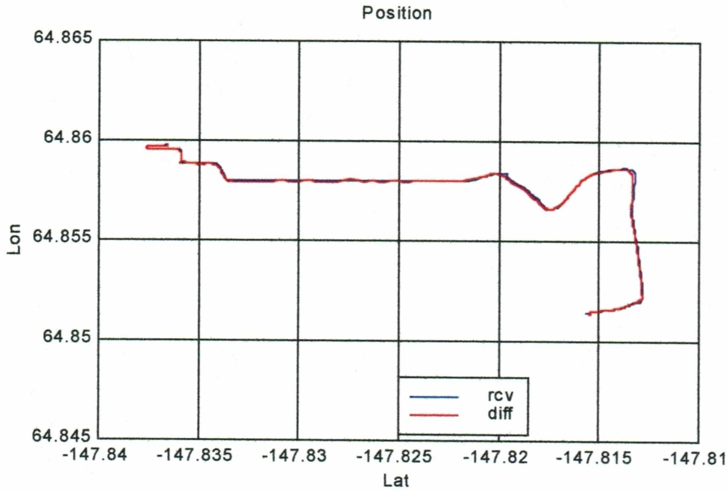

Figure 47: Trip to UPark, Position

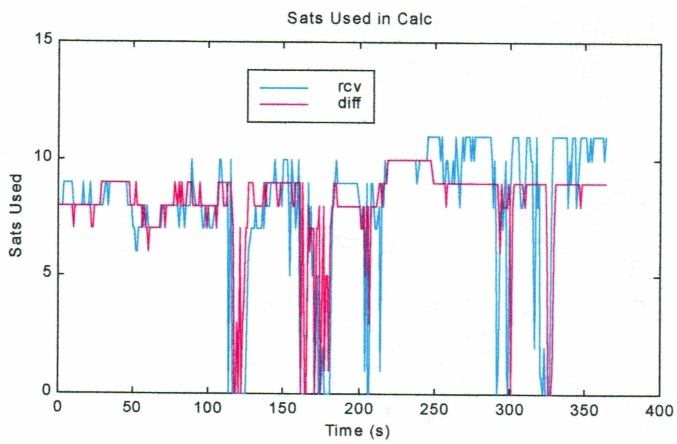

Figure 48: Trip to UPark, Satellites Used

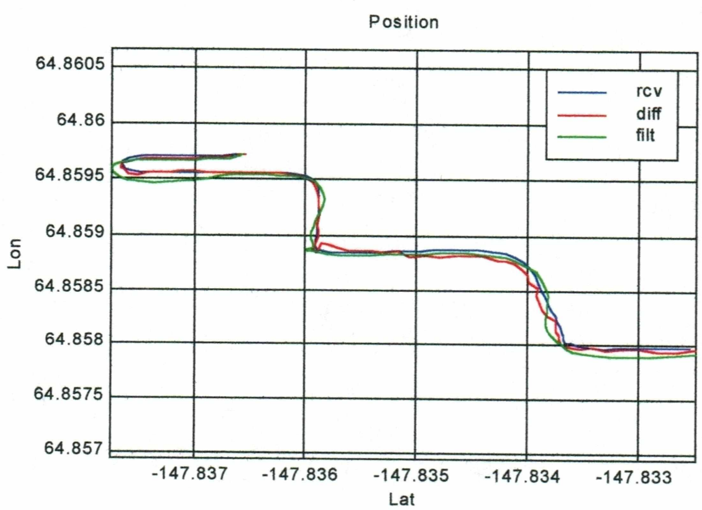

Figure 49: Trip to UPark, Going Past Natural Sciences Building

Chapter 6: Flight Experience

6.1 Sub-SEM

The GPS receiver flew on a NASA Suborbital Student Experiment Module (Sub-SEM) flight in June of 2000. This was an Orion vehicle that was launched from Wallops Flight Facility in Virginia. Two Micropulse 12700 antennas were used. They were mounted at opposing sides of the rocket on the Wallops Support Module (WSM) and combined using a power splitter. This layout provided continuous coverage as the rocket rotated without a wraparound patch antenna, but has the disadvantage of causing power fluctuation due to the antenna gain changing as the rocket rotates. Figure 50 shows the payload section of the rocket on the vibration table. The red module is the WSM and the gold section right below that is the antenna and umbilical section that goes with it. The two GPS antennas are the white modules that can be seen on either side. They are mounted in the space that the S-band antenna would have gone if the payload was set up with a telemetry link to the ground.

A re-rad system was included on the launcher to make sure that the GPS was locked up when the rocket was launched. The re-rad antenna is shown in Figure 52, the black antenna on the launch rail that is pointed at the white antenna on the rocket. An antenna with a clear view of the sky receives the satellite signals and they are re-transmitted from the re-rad antenna for the GPS antenna on the rocket to pickup. This is done because the rail blocks a large portion of the sky, making it hard for the on-board antenna to track many satellites by itself. A ground station was set up at the block house about 200 feet from the launcher. Figure 51 shows the ground station board and the laptop that recorded the data. On the day of the flight, the power cord didn't get plugged all the way in and the battery ran dead before the rocket was launched, so no differential corrections are available for the flight.

The flight board that was flown was the first prototype of the storage board. This

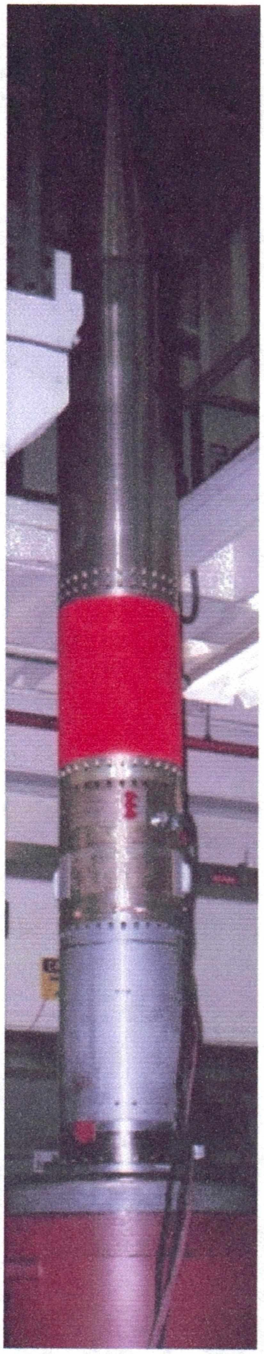

Figure 50: Payload on vibration table

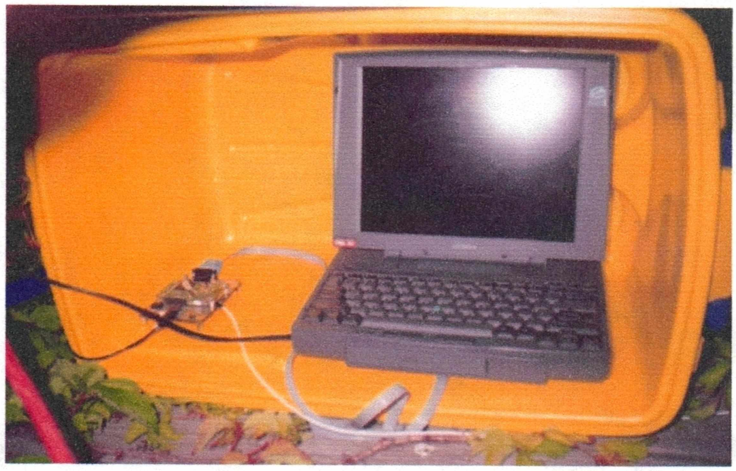

Figure 51: Ground station setup

design was used because there was no telemetry system on the rocket and all the data had to be stored on board. The jumpers on the board passed both the vibration testing and the subsequent flight without any sign of becoming loose. Based on this testing, jumpers are deemed flight qualified if the jumper cap is not obviously loose.

The flight was a partial success. The receiver tracked the satellites for one minute after launch. This time included the full

Figure 52: Re-rad antenna on the rail

thrust and burnout phase of the rocket motor. Unfortunately, the raw data was corrupted when the file was compressed and no processing can be done to verify that the receiver was tracking properly during this time. Due to this corruption, analysis information is based on memory of the data when it was first examined after the flight, and the few packets that are still good. No hardware problems were found on the ground and the receiver worked fine after the flight.

The radar data shows that the rocket was going through 31 km, or 100,000 ft at the one minute mark . It is possible that a software lockout tripped at this point. The receiver still output packets for the rest of the flight, but showed no signal strength on any of the channels. The U.S. export regulations require a software lockout at 1000 knots and 60,000 ft. Both of these limits were exceeded with the receiver apparently tracking normally. The speed limit was exceeded at 13.9 seconds, and the altitude limit was exceeded at 33.8 seconds, right after the motor burned out.

The MS-DOS® program is unable to process the few good data packets since they are so sporadic that the ephemeris cannot be extracted. Because of this, the raw data was examined by hand to try and get a feel for the quality of the data. Tables 8 and 9 show a comparison of the data in two adjoining packets before and after the launch. Table 8 shows that before launch there is a very good correlation between the new pseudorange that is output and the pseudorange that is predicted based on the old pseudorange and the rate of change. Table 9 shows that this continues to be the case after launch. This data set shows that the pseudoranges are apparently good even after no signal is being indicated on the different channels.

Table 8: Sub-SEM data comparison before launch

T-1min	sat id #	C/No	pseudorange (m)	range rate (m/s)	error from predicted (m)
	15	41	24978500.010	691.287	
	8	49	21005491.445	404.726	
	2	47	21664148.926	-407.742	
	27	45	21706038.960	553.278	
	7	45	22873033.252	-398.896	
	31	42	24325430.606	580.915	
	19	35	24525683.283	709.786	
	11	50	21092883.566	248.286	
	20	38	24487629.091	-526.386	
	26	35	25565617.477	196.663	
	13	38	25996429.679	470.946	
T-58 sec					
	15	41	24979882.595	691.248	-0.050
	8	49	21006300.926	404.834	0.079
	2	47	21663333.495	-407.713	-0.024
	27	45	21707145.645	553.308	-0.099
	7	45	22872235.574	-398.839	-0.057
	31	42	24326592.345	580.871	0.047
	19	36	24527100.985	709.835	1.919
	11	50	21093380.254	248.409	0.007
	20	38	24486576.438	-526.591	-0.324
	26	36	25566011.494	196.796	-0.558
	13	37	25997371.440	470.768	-0.047

The receiver was in operation during the vibration testing of the payload at Wallops and had no problem remaining locked up during the test. From this it is concluded that the small scale vibrations are not a problem to the receiver. It is expected that the most difficult times for the receiver to remain locked up and tracking well are during ignition and burnout. For the Orion motor there are two burnout transitions, the end of high thrust, and then the end of motor burn. The data collected from the flight shows that, though it can't be said that it positively tracked through these points, it did appear to do fine in the burnout transitions. It is speculated that the sudden dynamics during ignition caused the receiver to report a loss of lock because it dropped calculations in the second that the motor ignited. If just the software limits had taken effect it should have output a few packets after ignition before the speed limit was exceeded.

Table 9: Sub-SEM data comparison after launch

	sat id #	C/No	pseudorange (m)	range rate (m/s)	error from predicted (m)
T+59	15	42	25066183.288	813.522	
	8	45	21033419.435	227.198	
	2	0			
	27	44	21754971.129	390.861	
	7	43	22818335.358	-380.356	
	31	43	24389726.091	544.279	
	19	42	24604227.284	640.2	
	11	0			
	20	43	24414249.728	-658.102	
	26	42	25601382.517	455.647	
	13	41	26058913.658	599.717	
T+60	15	40	25066996.890	813.522	-0.080
	8	43	21033646.701	227.198	-0.068
	2	0			
	27	43	21755362.021	391.861	0.469
	7	41	22817955.080	-380.356	-0.078
	31	40	24390270.541	544.279	-0.171
	19	40	24604867.219	640.2	0.265
	11	0			
	20	40	24413591.373	-658.102	0.253
	26	40	25601838.184	455.647	-0.020
	13	39	26059513.384	599.717	-0.009
T+64	15	0	25070251.288	813.522	-0.310
	8	0	21034555.794	227.198	-0.301
	2	0			
	27	0	21756925.559	390.861	1.906
	7	0	22816435.661	-379.837	-0.967
	31	0	24392448.328	544.279	-0.671
	19	0	24607426.961	640.2	1.058
	11	0			
	20	0	24410957.926	-658.102	1.039
	26	0	25603660.823	455.647	-0.051
	13	0	26061912.309	599.717	-0.057

One possible explanation of the poor tracking result is that the antenna pattern variation caused too much fluctuation in the signals for the receiver to track properly. During testing of the Storage system with a similar antenna setup it was discovered that the receiver does not track very well when the payload is being rotated quickly. This would explain the drop after ignition because the Orion motor accelerates and spins up rapidly. However, this doesn't clearly explain why it still was able to track for a minute before completely losing all the satellites.

6.2 Hybrid

A full payload has been built but it has not been successfully launched yet. Ground testing with the payload has shown that the GPS receiver must be more than a foot from the antenna or well shielded to prevent the loss of received signal strength through some feedback mechanism. Spin tests with the payload have also shown that the receiver has trouble tracking the satellites while spinning with two side mounted antennas.

Chapter 7: Conclusions

All the hardware and software necessary for this thesis has been successfully constructed and developed. This includes the ground station support and two different flight systems. The two flight systems allow the GPS to be used with or without telemetry.

The software program works as designed. The quality of the output is highly dependent on the quality of the input data. This is particularly true relating to the differential corrections. With high quality data, the filtered position is within 2.2 feet of the correct location 90% of the time. With lower quality, data the filtered position is within 17.7 feet 90% of the time. The dual antenna setup shows that errors that are not correlated between the receivers are passed through the calculation. But even in this case, the location is still re-centered. Even if the relative error bars do not diminish much, the absolute error is reduced by having the offset removed.

The non-differential solution, if variance limited or filtered, is of similar accuracy to the receiver's calculations, which validates the software algorithm developed for this thesis. There is a long term drift on the order of 30 feet, and a short term drift of 10 to 20 feet depending on how data is filtered. If ground data is not available on a flight where the software limits were exceeded, good position information can still be obtained.

Both calculated data sets are not filtered as well or as heavily as the receiver's output so that it jumps around more. This could be smoothed out by improving the Kalman filter in the software or by smoothing and/or interpolating the data output by the program.

The Sub-SEM flight was partially successful. The receiver tracked well through the thrust and burnout phase of the launch before cutting out at 100,000 ft. The Rockwell Jupiter receiver was chosen because it is supposed to only have the software lockouts on the calculated position and not the raw data. This appeared to be the case for the first minute, as during this time both the speed and altitude limit had been surpassed. It is unknown whether this was a hardware or software problem. The evidence points more to a software lockout, but it seems very strange that there would be a lockout that was not consistent with U.S. export regulations. Since the receiver did not start operating again after dropping out, it is possible that the receiver must be reset before it will resume. It is also possible that the receiver was still tracking the satellites, but was just indicating that it wasn't. Another flight is need to make conclusive evaluations.

Appendix A: Hardware Schematics

A.1 Telemetry Flight Board

The latest version of this board is called 'flt gps'.

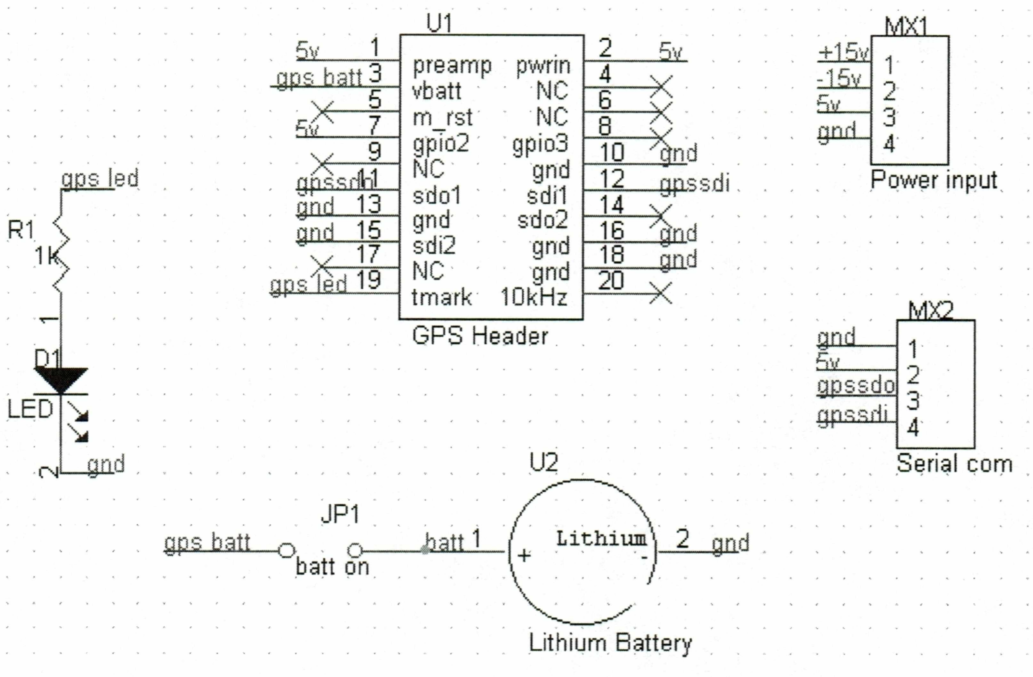

Figure 53: Detailed Schematic of Telemetry Flight Board

A.2 Storage Flight Board

The latest version of this board is called 'hybrid v4'.

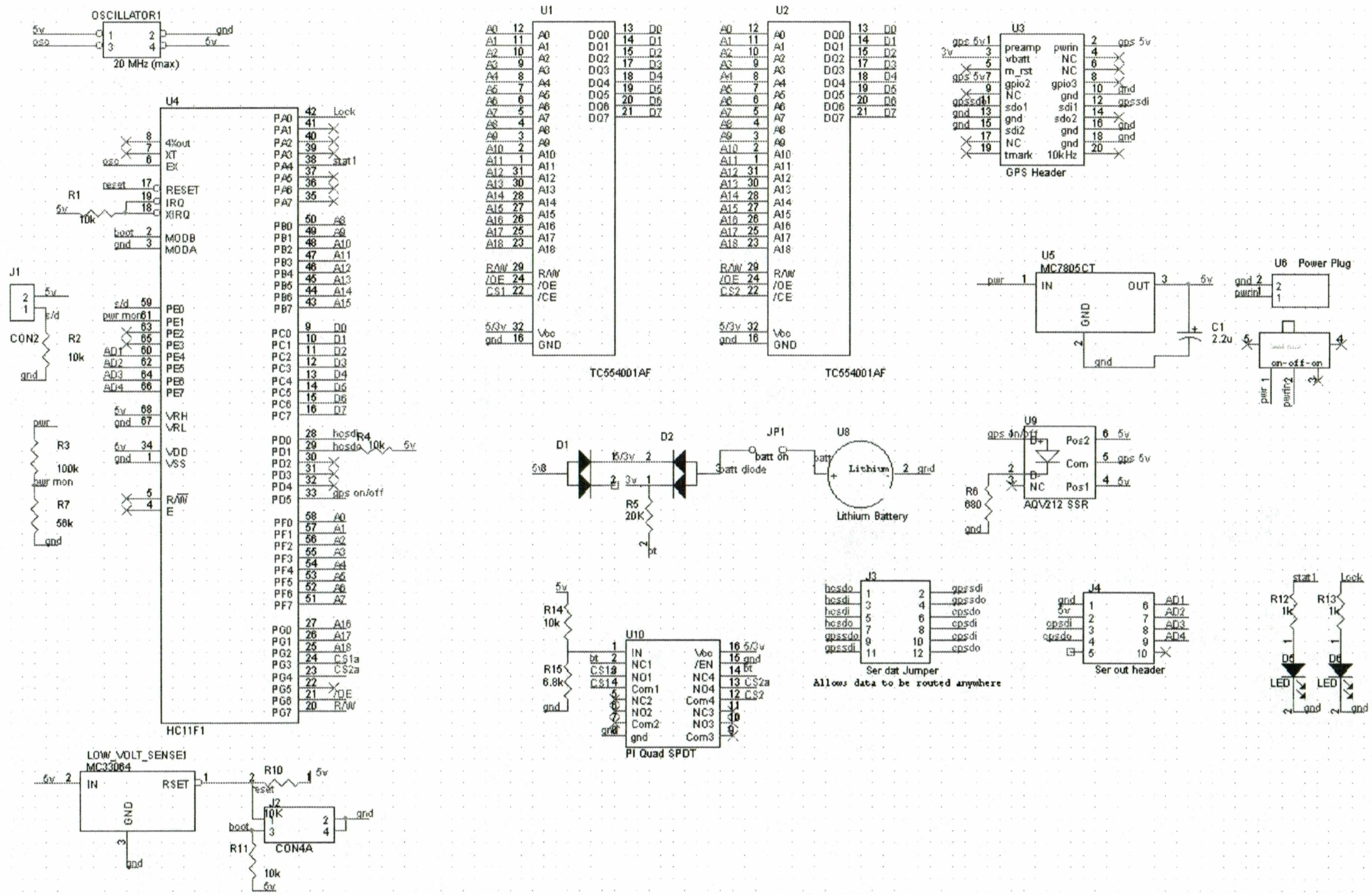

Figure 54: Detailed Schematic of Storage Flight Board

A.3 Ground Station Board

The latest version of this board is called ‘gnd gps 3’.

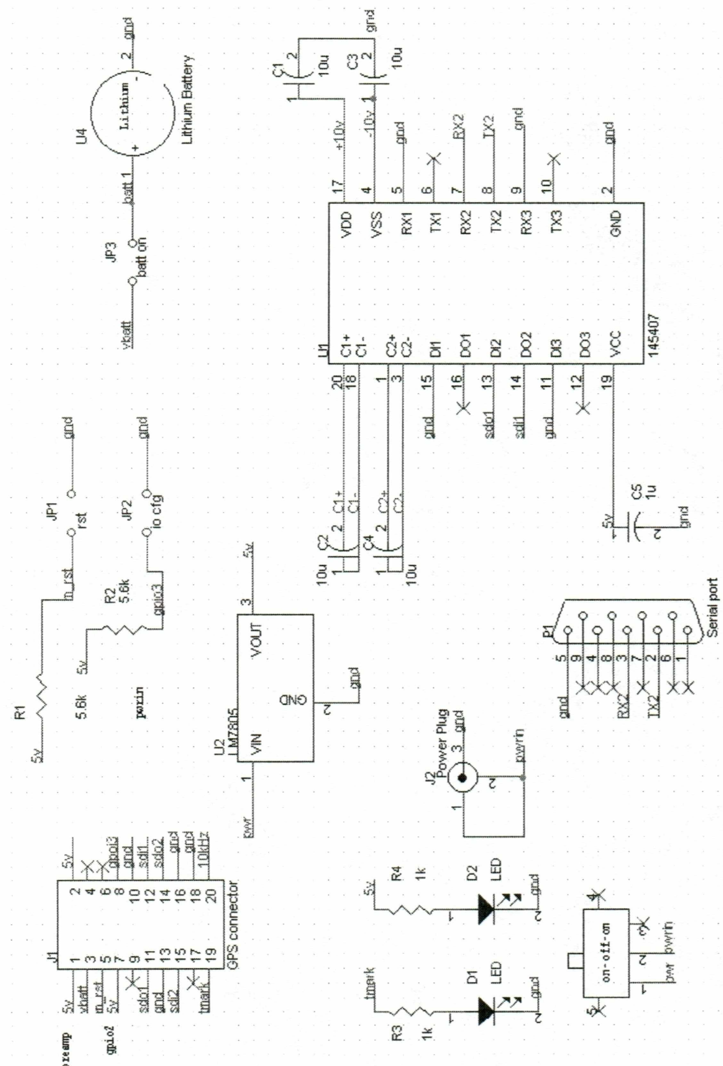

Figure 55: Detailed Schematic of Ground Station Board

Appendix B: Receiver Packet Formats

Table 10 lists the type abbreviations that are used in the packet listings.

Table 10: Data Types

Type	Abbreviation	bits	Type	Abbreviation	bits
Bit	Bit	0 to 15	Triple integer	TI	48
Character	C	8	Unsigned I	UI	16
Integer	I	16	Unsigned DI	UDI	32
Double integer	DI	32	Unsigned TI	UTI	48

All the packets have the following header format. Word 6-9 are only present if the header is a message log control command

Table 11: Message Header Format

word	name	info
1	synch. And start of header	always 10000001 11111111
2	Message id	binary representation of mesage id
3	data word count	number of words in data portion of message not including checksum
4	DCL0 QRAN 00xx xxxx	one means that it is set
5	checksum	twos complement of sum of four header words
word	**name**	**info**
6	Trigger	0=on time, 1=on update
7	interval	in sec between updates if time is selected 0 to 65535
8	Offset	in sec before message is sent 0 to 60
9	data checksum	Two's complement of sum of three message words

Word 4 holds the protocol and message flags.

D = disconnect, or disable the message in word 2.

C = connect, or enable the message in word 2.

L = will start logging the message in word 2. If the default timing is not changed than the extra four words are not needed.

Q = query bit sends a one time output of the message in 2

R = request bit for A or N

A = acknowledge successful requests

N = negative acknowledge, responds only if there is a problem with the request.

xxxxxx = id number that will be used in the reply.

The three zero's must be zero.

Packet 1000 is the basic navigation packet. Word 10 is the navigation solution validity and word 11 is the navigation solution type.

Table 12: Detailed Packet Format for Message 1000

Packet	**1000**		**55 words**
word	**name**	**type**	**info.**
1-4	Header		
5	Header Checksum		Two's complement of sum of header
6-7	Set time	udi	10 ms count since power up, provides sequence of events
8	sequence number	I	sequence number increments every time the data is updated.
9	satellite measurement sequence number	i	relates the position solution data to a particular set of satellite measurements found in 1002 &1007. tells what sats were used?
10.0	solution invalid-altitude used	bit	1 = true
10.1	solution invalid-No differential	bit	1 = true
10.2	Not enough satellites	bit	1 = true
10.3	Exceeded Max EHPE	bit	1 = true ehpe = est. horisontal position error
10.4	Exceeded Max EVPE	bit	1 = true evpe = est. virtical position error
10.5-10.15	reserved		
11.0	Propagated	bit	1 = true message was output by I/O manager without new data
11.1	altitude used	bit	1 = true
11.2	differential	bit	1 = true
11.3-11.15	reserved		
12	# of measurements in soulution	ui	0 to 12
13	polar Navigaion	bit	1 = true
14	gps week #	ui	

15-16	gps seconds from epoch	udi	
17-18	gps nanoseconds form epoch	udi	
19	utc day	ui	
20	utc month	ui	
21	utc year	ui	
22	utc hour	ui	
23	utc minutes	ui	
24	utc second	ui	
25-26	utc nanosecond	udi	
27-28	Latitude	udi	rads +- 0 to pi/2
29-30	Longitude	udi	rads +- 0 to pi
31-32	Height	udi	meters
33	Geoidal Separation	i	meters
34-35	Ground speed	udi	m/s
36	True course	ui	rads 0 to 2pi
37	magnetic variation	i	rads, add to true course to get magnitic course
38	climb rate	i	m/s
39	map datum	ui	0-188 Normally 0-WGS84 121-North America may be more acurate
			Est. errors are not valiad unless navigating.
40-41	ehpe	udi	meters est. hoizontal position error
42-43	evpe	udi	meters est. vertical position error
44-45	ete	udi	meters est. time error
46	ehve	ui	m/s est. horizontal velocity error
47-48	clock bias	di	meters
49-50	clock bias standard devation	di	meters
51-52	Clock drift	di	m/s
53-54	Clock drift standard devation	di	m/s
55	data checksum		Two's complement of sum of data

Packet 1003 lists the visible satellites and their positions.

Table 13: Detailed Packet Format for Message 1003

Packet	**1003**		**51 words**
word	**name**	**type**	**info.**
1-4	Header		
5	Header Checksum		Two's complement of sum of header
6-7	Set time	udi	10 ms count since power up, provides sequence of events
8	sequence number	I	sequence number increments every time the

			data is updated.
9	Best Possible GDOP	i	geometric
10	Best Possible PDOP	i	position
11	Best Possible HDOP	i	horizontal
12	Best Possible VDOP	i	vertical
13	Best Possible TDOP	i	time
14	# of visible satellites	ui	1 to 12
	Visible satellite set j goes from 0 to 11		
15+3*j	Satellite PRN	ui	PRN is pseudo random noise #, just the id # of satellite
16+3*j	Satellite Azimuth	i	
17+3*j	Satellite Elevation	i	
51	data checksum		Two's complement of sum of data

Packet 1102 is the raw data output packet.

Table 14: Detailed Packet Format for Message 1102

Packet	**1102**		**253 words**
word	**name**	**type**	**info.**
1-4	Header		
5	Header Checksum		Two's complement of sum of header
6-7	Set time	udi	10 ms count since power up, provides sequence of events
8	sequence number	I	sequence number increments every time the data is updated.
9-12	gps measurement time		
	integer portion	di	sec. 0 to 604799.98 from start of week
	fractional portion	di	sec 0 to +- 0.2
13.0-13.1	reserved		
13.2	hand over word decoded	bit	1=set measurement engine has decoded at least on hand over word
13.3-13.15	reserved		
14-24	reserved		
n	data word subframe index		n=25+(j*19) where j=0 to 11
(n+1).0	weak signal	bit	1=signal strength fell below threshold
(n+1).1	High delta theta	bit	1=carrier phase change exceeded threshold
(n+1).2	Parity Errors	bit	1=carrier cycle slips may have affected this measurement or previous
(n+1).3	Reserved		

(n+1).4	Reserved		
(n+1).5	Bit sync flag	bit	1=unknown
(n+1).6	frame sync flag	bit	1=unknown
(n+1).7	Z Count flag	bit	1=unknown
(n+1).8-(n+1).15	reserved		
(n+2).0-(n+2).4	Pre-Detection interval	ui	1 to 20
(n+2).5-(n+2).15	reserved		
n+3	Satellite PRN		PRN is pseudo random noise #, just the id # of satellite
n+4	C/No		carrier to noise ratio dBHz
n+5	Code Phase Measurement	uti	pseudorange is #*c(2-45/50)
n+8	Carrier Phase Measurement	uti	continuously integrated carrier phase at the epoch
n+11	Carrier Velocity Measurement	di	
n+13	Code Phase Standare Deviation	ui	
n+14	Carrier Phase standard deviation	ui	
(n+15).0-(n+15).29	sv data word one		30bit subframe data word from satellite telemetry
(n+15).30	validity		1=valid
(n+15).31	Parity Error		1=error
(n+17).0-(n+17).29	sv data word two		30bit subframe data word from satellite telemetry
(n+17).30	validity		1=valid
(n+17).31	Parity error		1=error
253	data checksum		Two's complement of sum of data

Bibliography

Burket III, V. Edward (2001). *A Modular Electrical System Design for Sounding Rocket Payloads*. Manuscript. Available from the ASGP website: http://www.uaf.edu/asgp/

Dana, Peter H. (2001). *Global Positioning System Overview*. Available from: http://www.colorado.Edu/geography/gcraft/notes/gps/gps_f.html.

Gilbert, Chuck (2001). *Attribute Collection with a GPS.* Available from: http://www-aadc.antdiv.gov.au/mapping/gps/gps_series/partw.asp.

Kaplan, Elliott D., editor (1996). *Understanding GPS, Principles and Applications*. Artech House, Boston London.

Kleusberg, Alfred & Teunisen, Peter, editors (1996). *GPS for Geodesy*. Number 60 in Lecture Notes in Earth Sciences. Springer Verlag, Heidelberg.

Leick, Alfred (1995). *GPS Satellite Surveying*. John Wiley and Sons, Inc., New York Chichester Brisbane Toronto Singapore, 2nd edition.

NAVSTAR (1995). *Global Positioning System Standard Positioning Service Signal Specification*. 2nd edition. Available from the U.S. Coast Guard website: http://www.navcen.uscg.gov/pubs/gps/sigspec/default.htm.

Rockwell (1996). *Zodiac GPS Receiver Family Designer's Guide*. Rockwell International Corporation.

Sam (2001). *Smoothing of Pseudorange with Carrier Phase*. Available from: http://home-2.worldonline.nl/~samsv1/smooth.htm.

Strang, Gilbert & Kai Borre (1997). *Linear Algebra, Geodesy, and GPS.* Wellesley-Cambridge Press, Wellesley MA.